Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Multi-scale techniques for multi-dimensional data analysis

by

Nikolas De Giorgis

Theses Series **DIBRIS-TH-2018-03**

DIBRIS, Università di Genova
Via Opera Pia, 13 16145 Genova, Italy http://www.dibris.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica, Bioingegneria,

Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in Computer Science and Systems Engineering
Computer Science Curriculum

Multi-scale techniques for multi-dimensional data analysis

by

Nikolas De Giorgis

May, 2018

Dottorato di Ricerca in Informatica ed Ingegneria dei Sistemi
Indirizzo Informatica
1Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università degli Studi di Genova

DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy
http://www.dibris.unige.it/

Ph.D. Thesis in Computer Science and Systems Engineering
Computer Science Curriculum
(S.S.D. INF/01)

Submitted by ...
DIBRIS, Univ. di Genova
. . . .

Date of submission: January 2018

Title: Scale-space techniques for multi-dimensional data analysis

Advisor: Prof. Enrico Puppo
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
. . .

Ext. Reviewers:
Riccardo Scateni, Università di Cagliari
Silvia Mabel Castro, Universidad Nacional de Sur, Bahia Blanca

Abstract

Large datasets of geometric data of various nature are becoming more and more available as sensors become cheaper and more widely used. Due to both their size and their noisy nature, special techniques must be employed to deal with them correctly. In order to efficiently handle this amount of data and to tackle the technical challenges they pose, we propose techniques that analyze a scalar signal by means of its critical points (i.e. maxima and minima), ranking them on a scale of importance, by which we can extrapolate important information of the input signal separating it from noise, thus dramatically reducing the complexity of the problem.

In order to obtain a ranking of critical points we employ multi-scale techniques. The standard scale-space approach, however, is not sufficient when trying to track critical points across various scales. We start from an implementation of the scale-space which computes a linear interpolation between scales in order to make tracking of critical points easier. The linear interpolation of a process which is not itself linear, though, does not fulfill some theoretical properties of scale-space, thus making the tracking of critical points much harder. We propose an extension of this piecewise-linear scale-space implementation, which recovers the theoretical properties (e.g., to avoid the generation of new critical points as the scale increases) and keeps the tracking consistent. Next we combine the scale-space with another technique that comes from the topology theory: the classification of critical points based on their persistence value. While the scale-space applies a filtering in the frequency domain, by progressively smoothing the input signal with low-pass filters of increasing size, the computation of the persistence can be seen as a filtering applied in the amplitude domain, which progressively removes pairs of critical points based on their difference in amplitude. The two techniques, while being both relevant to the concept of scale, express different qualities of the critical points of the input signal; depending on the application domain we can use either of them, or, since they both have non-zero values only at critical points, they can be used together with a linear combination.

The thesis will be structured as follows: In Chapter 1 we will present an overview on the problem of analyzing huge geometric datasets, focusing on the problem of

dealing with their size and noise, and of reducing the problem to a subset of relevant samples. The Chapter 2 will contain a study of the state of the art in scale-space algorithms, followed by a more in-depth analysis of the virtually continuous framework used as base technique will be presented. In its last part, we will propose methods to extend these techniques in order to satisfy the axioms present in the continuous version of the scale-space and to have a stronger and more reliable tracking of critical points across scales, and the extraction of the persistence of critical points of a signal as a variant to the standard scale-space approach; we will show the differences between the two and discuss how to combine them. The Chapter 3 will introduce an ever growing source of data, the motion capture systems; we will motivate its importance by discussing the many applications in which it has been used for the past two decades. We will briefly summarize the different systems existing and then we will focus on a particular one, discussing its peculiarities and its output data. In Chapter 4, we will discuss the problem of studying intra-personal synchronization computed on data coming from such motion-capture systems. We will show how multi-scale approaches can be used to identify relevant instants in the motion and how these instants can be used to precisely study synchronization between the different parts of the body from which they are extracted. We will apply these techniques to the problem of generating a classifier to discriminate between martial artists of different skills who have been recorded doing karate's movements. In Chapter 5 will present a work on the automatic detection of relevant points of the human face from 3D data. We will show that the Gaussian curvature of the 3D surface is a good feature to distinguish the so-called fiducial points, but also that multi-scale techniques must be used to extract only relevant points and get rid of the noise.

In closing, Chapter 6 will discuss an ongoing work about motion segmentation; after an introduction about the meaning and different possibilities of motion segmentation we will present the data we work with, the approach used to identify segments and some preliminary tools and results.

Alla mia famiglia. Di ieri, oggi, e di domani.

The strain of anti-intellectualism has been a constant thread winding its way through our political and cultural life, nurtured by the false notion that democracy means that "my ignorance is just as good as your knowledge" (Isaac Asimov)

Acknowledgements

Diventa sempre più difficile ringraziare tutte le persone che se lo meritano. Passano gli anni e siete sempre di più ad essermi stati vicini, ad avermi aiutato nei momenti più difficili, ad avermi dato la forza, ad avermi fatto ridere, sorridere, anche semplicemente ad avermi fatto dimenticare per un attimo preoccupazioni, difficoltà, fatica. Se oggi sono la persona che sono, e se posso dirmi fiero di dove sono arrivato, è merito di tutti voi.

I miei genitori saranno sempre qui, in cima alla lista. Questo momento è tutto grazie a voi, che avete sempre creduto in me, che mi avete supportato per tutti questi anni, che mi siete stati sempre vicini e che mi avete donato tutti i vostri insegnamenti. I ringraziamenti per voi non saranno mai abbastanza, e continuerò ad esservi grato per il resto della mia vita. E con voi anche Lavinia, ed il nostro rapporto non sempre facile, ma fatto di vero amore fraterno nel momento del bisogno. E ovviamente Greta, con tutto il suo entusiasmo la sua iperattività, l'amore incondizionato che è in grado di dare, gli abbracci dopo settimane senza vederci.

Camilla, per quanto la vita possa riservarci difficoltà, scontri, litigi, per quanto possa provare a mettere km di distanza tra noi due, dentro di noi non si spegnerà mai la convinzione che il nostro futuro sia insieme. Quando c'è l'intesa, la complicità, l'entusiasmo che ancora oggi, dal primo giorno, proviamo, tutto il resto sparisce, e rimane solo la visione di una vita insieme, sempre felici come il primo giorno. In questi anni sei stata la luce nei momenti più bui, l'unica persona in grado di farmi dimenticare in un secondo tutti i problemi, gli affanni, le difficoltà. Senza di te non so dove sarei, ma so che non sarei felice come lo sono ora.

All'amico di una vita Giorgio. Anche se abbiamo accettato il fatto di essere (quasi) delle persone serie e dover avere una vita, siamo sempre due deficienti con la battuta (che fa ridere solo noi) pronta. Anzi, spesso la battuta non dobbiamo neanche farla, basta guardarci e già l'abbiamo capita e stiamo ridendo. Vorrei scrivere qualcosa di più sulla nostra amicizia e su quanto sia stata importante in questi 18 anni che ci conosciamo (DICIOTTO! SONO TANTISSIMI!) ma so che diventerei troppo emotivo e me lo rinfacceresti per il resto della mia vita.

Alle amicizie nate tra i tavoli del quinto piano e proseguite al settimo: a Luca e alle pesantissime cene prima di giocare a calcio, alla Vero e alla sua predisposizione a far notare la rotondità del mio girovita, a Susi che mi ha fatto rendere conto di non saper ridere, a Pietro ed i giri per Nervi le poche volte che si riesce a schiodarlo da casa. A chi mi ha fatto sentire meno solo quando vedevo gli amici di 5 anni di università piano piano andarsene mentre io rimanevo lì: a Dasse, Prampo, Fra Rosasco, allo scopone. Grazie a Dimi, che mi ha fatto capire che ogni tanto la prima impressione su una persona può anche ingannarmi.

Un grazie a tutti coloro che hanno reso la 309 un posto un po' meno cupo e lugubre: a Corradi, che non sopportandomi più ha pensato bene di dottorarsi in tempo; alla ritrovata Piccia, ad Angelo che invece mi ha dovuto sopportare parecchio, a Fede, Samu, Muccia, Chiara, Vane (ma che bella donna la D'Amario!), a Fra Dagnino e al fiume di parole che risponde al nome di Laura. Grazie a chi in 309 passava da foresto ma sempre gradito: grazie a Dama e Gigi.

Grazie a chi ha tenuto i contatti anche dopo il capitolo dell'università, nonostante la vita vada avanti e cerchi di sfilacciare i contatti tanto duramente intrecciati: Sonia, dovunque tu vada sarai sempre un'amica fidata, una delle poche persone in grado di farmi ridere SEMPRE. Grazie Pasto, anche se siamo due teste calde siamo riusciti a scontrarci un numero piuttosto limitato di volte, in questi TOT anni di amicizia (ho rinunciato a contarli, non sono proprio in grado). Grazie a Luca Turchi, sempre bellissimo. E grazie a quei pochi, ma buonissimi, che dai tempi delle superiori sono rimasti amici nonostante le nostre vite tendano ad allontanarci: a Dago, il Biondo, Sten, alla londinese Emma, e a colei che coniò il soprannome con cui tutti ormai mi conoscono, Sofia (no, non è Sofia il soprannome).

Se sono sopravvissuto tutti questi anni senza impazzire (non so quanto questa affermazione sia vera, in realtà), devo molto allo sfogo settimanale del calcetto; un modo per buttare via tutta la tensione e la rabbia che inevitabilmente si accumulano durante la settimana, con risate, pallonate e, perché no, qualche infortunio (principalmente mio). Grazie a tutti quelli con cui ho condiviso il campetto più e più volte; grazie a Raffa, Stocco, ad Asan ,a Giallo, Joan, Ulde, Bandi.

Grazie agli amici di Ingress. Per qualcuno sarà un gioco da sfigati (beh dai, un po' lo è, lo sappiamo anche noi) ma per me è stata l'occasione di conoscere gente meravigliosa con cui condividere momenti fantastici, che fosse salire alle 5 di notte con la temperatura sotto zero a Forte Geremia (ma fornitissimi di caffè!) con Ezio, o soffrire la folle guida di Alex per stradine che non saprei mai più ritrovare, da qualche parte vicino a Tiglieto, o scivolare giù dal Cavalmurone con le palette di Biste. Grazie a tutti voi, grazie Bea, Dario, Camm, LL, Holy, Franz, Tangi, Gemi, Loppe. E perché no, grazie anche a qualcuno dell'altra squadra: grazie Fede, Filo, Andry. E anche se molti di voi li ho incontrati di persona solo una volta per pochi minuti, grazie al pazzo mondo dei decoder: Kuprum (BESTIA), Awleps, Corvo, DD, e, al netto di litigi e ragequit, pure a sooshee e ifonz. Anche a tutti gli stranieri, ma non avrò mai voglia di mettermi a tradurre tutto.

È sempre difficile finire i ringraziamenti. Il vuoto del foglio bianco ti fa capire che sicuramente hai dimenticato qualcuno, che molta gente meritava di più, che molti, moltissimi dei citati non sarai mai in grado di ringraziarli abbastanza. Ma quella sensazione di caldo al cuore mentre scorri la lista ti fa anche capire che la vera magia di questi anni insieme sta nel non aver bisogno di tutti questi sproloqui per spiegarci cosa vuol dire l'amicizia. Se vi ho dimenticato, se non vi ho ringraziato quanto meritavate, sappiate che quando ci vedremo vi offrirò una birra e sarà un'altra, l'ennesima, bellissima serata tra amici.

Table of Contents

Chapter 1 Representation and manipulation of geometric datasets 4

1.1 Geometric datasets 4

1.2 Manipulation of geometric datasets 5

1.3 Analysis of geometric datasets 7

1.3.1 Analysis of scalar fields 8

Chapter 2 Dealing with scale 11

2.1 Introduction 11

2.2 Scale-Space Analysis 12

2.2.1 Limitations of the linear approximation 14

2.2.2 Conclusions 19

2.3 Persistence 19

2.3.1 Conclusions 23

2.4 Combining the multi-scale analysis 23

Chapter 3 Motion Capture Data 26

3.1 Introduction 26

3.2 Applications 27

3.2.1 Animation database 27

3.3 Acquisition Process 28

3.3.1 Non-Optical System 28

3.3.2 Optical System 28
3.4 Data 31
3.5 Analysis 31

Chapter 4 Analysis Of Synchronization from Karate's motion capture recordings 33
4.1 Introduction 34
4.2 Related Work 36
4.3 Extraction of events 36
4.4 Synchronization 37
4.5 Results 41
4.5.1 Parameters setting 41
4.5.2 Statistical analysis 43
4.5.3 A basic classifier 47
4.6 Conclusion and Future work 48

Chapter 5 Segmentation Of Human Motion 50
5.1 Introduction 50
5.2 Datasets used 51
5.3 Feature selection 53
5.4 The concept of scale in motion segmentation 55
5.4.1 Space 55
5.4.2 Time 55
5.5 Extracting the segmentation 56
5.6 Preliminary Results 56
5.6.1 Analysis of ground truth data 57

Chapter 6 Scale-Space Techniques for Fiducial Points Extraction from 3D Faces 58
6.1 Introduction 58
6.2 Fiducial Points 59

6.3 State Of The Art for Feature Based Techniques 60
6.4 Surface Curvature 61
6.4.1 Background 61
6.5 Fiducial Points Characterization 63
6.6 Method 63
6.6.1 Curvature Extraction 63
6.6.2 Diagonal Scale-Space 64
6.6.3 Critical Points' Importance Measure 66
6.6.4 Extraction of fiducial points 66
6.6.5 Results 70

Chapter 7 Conclusions and Future Work **73**
7.1 Future Works 74

Publications **75**

List of Figures **76**

Bibliography **79**

Chapter 1

Representation and manipulation of geometric datasets

This chapter will present the context of the data on which we worked with our methods. It will give a general description of Geometric datasets, with their distinguishing features, the various ways in which they can be stored and represented and their mathematical properties. A summary of the issues related to the analysis of geometric datasets of different nature follows, together with a description of the properties that we seek when developing techniques aimed at their analysis.

1.1 Geometric datasets

Geometric datasets can be generically seen as a way to describe an object; there are various criteria that can characterize an object:

- Its **dimensionality and embedding**: The dimension of an object can be intuitively seen as the number of coordinates needed to unambiguously identify a point inside of it; a line is a *one-dimensional* entity, a plane is two dimensional, a cube is represented by at least three coordinates; higher dimensions come into play when other qualities are considered: for example, to describe the point of a cube moving through time, four coordinates are needed. The embedding is the dimension of the space the object is immersed into: a two-dimensional line that is, for example, the diagonal of a cube, is one-dimensional, but to define it we need a 3D space. In the rest of the thesis, unless differently specified, we will deal with objects defined with dimension $\mathbb{R}^n$ and embedding in $\mathbb{R}^m$ (i.e. both the spaces are defined by real coordinates) . It is intuitive to see that when an object of dimension $\mathbb{R}^n$ is embedded in $\mathbb{R}^m$, the relation $n \leq m$ holds.

- Its **connectivity**, i.e. the organization of the parts composing it into some kind of structure. If there is no structure at all we have an unorganized collection of coordinates, usually referred to as a *point cloud*; an intuitive and basic kind of structure are *lattices*, i.e. equally spaced array of points (e.g. points defined on a regular 2D grid, such as pixels of an image), while more complex ones such as *graphs* allow for more complex structures. The most common *graph-based* representations of geometric datasets are *meshes*.

There are other criteria which might give other characterizations of geometric datasets, such as *continuous* vs *discrete* representation, but since in this thesis the most common differences between different domains will be in the two aforementioned categories we will focus on those ones.

Geometric datasets can be enriched by associating to each point additional values, such as scalar values or vectors, or a combination of them. We will call them *scalar fields*, *vector* and *tensor* fields respectively. It is worth noting that these fields, when defined on datasets in which some concept of connectivity is present, allow for interpolation by exploiting the connectivity between points.

In this thesis , we will work mostly one and two dimensional datasets, embedded in two, three or four dimensional spaces, on which scalar fields are defined; the techniques presented in the thesis have been developed to work with this kind of data, but these could be extended higher dimensionality fields present in different domains.

1.2 Manipulation of geometric datasets

The analysis of geometric datasets poses numerous challenges, as there are intrinsic characteristics, other than the one described in the previous section, that introduce variability and both theoretical and computational issues.

- **Resolution**: sensors' manufacturers are constantly increasing the resolution at which their products can work; high resolution 3D meshes can be obtained with relatively cheap equipment, e.g., a 3D model of a face can be obtained with a reflex camera and software under 100$; while motion capture systems, albeit being much more expensive, are increasing sampling's frequency to 300Hz and even higher. Then, objects in geometric datasets tend now to be represented at very high resolutions, which has two main consequences:
  - the **size** can escalate quickly, thus making classic algorithms computationally prohibitive: for example, a two minutes long motion capture recording, sampled at 250Hz, will consist of 30,000 triples of (x, y, z) coordinates for each sampled marker; the increasing length of the recordings and the potential addition features to be recorded

(e.g., a vector field associated to each frame) will lead very quickly to enormous datasets. One might be tempted to solve the problem by simply downsampling the signal to a more manageable one, thus risking of losing information that is needed to describe the represented object (e.g. very fast human movement might need at least 100hz frequency)

  - the **scale** of the structures contained in the represented objects: a higher resolution in the representation means that it is potentially possible to identifyis a wider range of structures; this means that methods are needed to distinguish between features at different scale. This concept can be expressed with a classic analogy: if we look at a forest from far away we just see a green spot and we can not distinguish other features; but if we look at a different *scale* (i.e., we zoom in) we can start to tell different trees apart from each other; by continuing this process we can then see each branch, and eventually we can see each leaf separately. The same situation arises when analyzing different signals: there might be relevant information at various levels of detail, from the coarsest one to the finest one, which is usually the one at which noise and signal become indistinguishable. In order to capture all the information a signal can carry, most of the literature since the 80s has been developed around the concept of *scale-space* and its related techniques, introduced by the seminal works of Witkin [Wit83] and Koenderink [Koe84]. This representation has then been enriched with multiple extensions and concepts, especially the *deep structure*, i.e., the zero crossings of some differential invariants defined on the signal [Lin94].

- the **noise** present in the representation has to be recognized and separated from the actual information the signal carries: this problem is always present when analyzing signals coming from the real world, and gets even bigger with the increased availability of cheap and fast sensors, that might sacrifice the precision of the recorded signal to be faster, and where processing of the raw input is done by the sensors themselves, freeing the user to do any post-processing and data cleaning. Taking into consideration also the fact that sensors might be used to record something that is inherently noisy, such as the human motion, it comes as a natural consideration that noise should be treated with particular care. There are a lot of different types of noise, which have been widely studied in the literature, specially in the signal processing field [Vas06, Unc16, Coh05]. In our case, we will restrict to the following types of noise:

  - *Additive*: is a kind of noise that is added to the original signal, it is white (i.e., it has uniform power across the frequency band of the information system) and it is of Gaussian type (a normal distribution with an average of zero). This is usually used in information theory to model the effect of random natural processes.
  - *Multiplicative*: is an unwanted random signal that gets multiplied into some relevant signal during the acquisition phase.
  - *Quantization*: is the difference between the original value and its quantized value.

1.3 Analysis of geometric datasets

Given all the considerations presented in the previous sections, the need for special methods to analyze this kind of data arises; such techniques should be able to:

- Deal with datasets of different type, from uni-dimensional datasets to time-varying 3D datasets, without loss of expressive power and where the overhead introduced is the minimum needed to extend their functionality to the various types of datasets;
- Analyze huge datasets in an efficient way: even though real-time processing of dataset is not often needed, there are always temporal constraints: in case of offline processing, the algorithms developed should be able to output a result in a reasonable time; furthermore, the output should be a comprehensive representation solving the posed problem , but also compact enough to be analyzable without further overhead;
- Be flexible with respect to the scale of the structures represented in the dataset: an ideal algorithm is one that makes no assumptions on what scale the objects live at, and is able to output a representation that carries information about each structure found in the input dataset, together with a quantitative measure of its *scale*;
- Be robust to the various kind of noise present in the dataset.

Our intent is to reduce the complexity of the problem by analyzing not the original signal itself, but a *scalar field* defined on it: suppose we have an object embedded in $\mathbb{R}^d$, we can have a scalar field $f : \mathbb{R}^d \longrightarrow \mathbb{R}$. Since our objects are represented discretely, the function f will be actually defined only for those points of $\mathbb{R}^d$ that corresponds to the discrete representation. By exploiting the concept of connectivity possibly present in our dataset, the function can be interpolated and its definition extended; the simplest example is given by linear interpolation:

- *Linear interpolation*: given two points x_1 and x_2, with associated values $f(x_1) = y_1$ and $f(x_2) = y_2$, the value of the function f can be interpolated along the straight line segment (x_1, x_2) with the following formula:

  $$f(x) \approx \frac{f(x_1)-f(x_0)}{x_1-x_0}(x - x_0).$$

  Linear interpolation can be defined for simplexes of every dimension; we give here the definition of linear interpolation on a 2-dimensional simplex, i.e. a triangle (this is the kind of linear interpolation that can be used on a scalar field defined on the vertices of a triangular mesh, as the one we will discuss about in chapter 6): given a triangle whose vertices are v_0, v_1 and v_2, a point p inside the triangle, and a function f defined on the three vertices, the interpolated value of f at point p is given by the following formula:

$$f(p) \approx af(v_0) + bf(v_1) + (1 - a - b)f(v_2)$$

with $a \geq 0$, $b \geq 0$ and $a+b <= 1$, where a, b and $(1-a-b)$ are the barycentric coordinates of p in the triangular domain.

- *Bilinear interpolation*: it interpolates a function of the variables on a quadrilateral 2D domain: assuming that we know the value of the function in four corner points of the domain (starting from the lower left corner and going clockwise: $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_1, y_2)$, $Q_{22} = (x_2, y_2)$, $Q_{21} = (x_2, y_y)$), we can interpolate for a given point (x, y), by first doing linear interpolation in one direction (for example, the x-direction):

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$
$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

and then by interpolating in the other direction

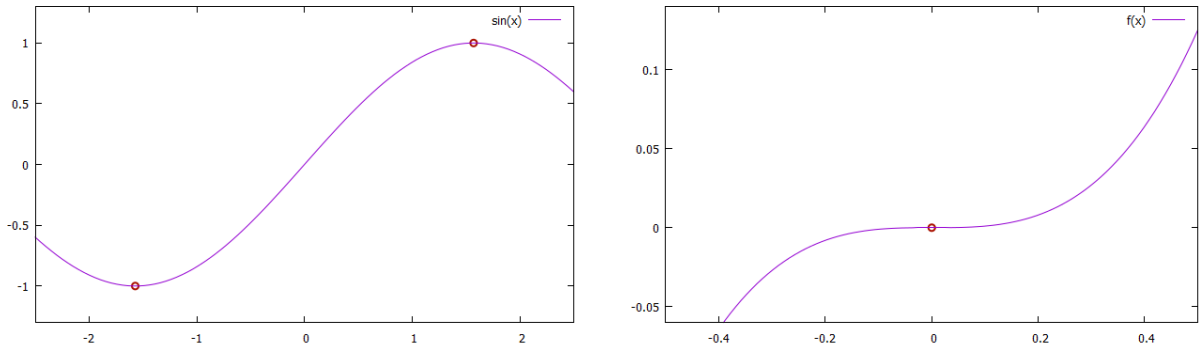
$$f(x, y) \approx \frac{(y_2 - y)}{y_2 - y_1} f(x, y_1) + \frac{(y - y_1)}{y_2 - y_1} f(x, y_2).$$

- *Trilinear interpolation*: it does the same as previous but for points lying on a 3-dimensional hexahedron. The formula is left out for the sake of brevity, but it works in the same way as the bilinear interpolation, i.e. by interpolating first on a dimension, than on the second one and finally on the third. The same approach can be actually extended to any dimension, provided that the domain of interpolation has the structure of a hypercube.

1.3.1 Analysis of scalar fields

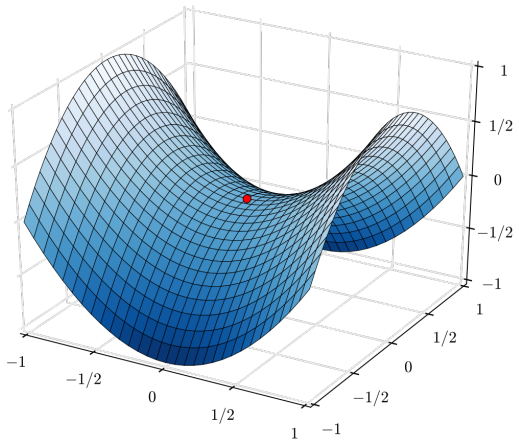
After applying one of the interpolation techniques described in the previous section (or similar), we now have a scalar field defined on the whole domain of the object represented in the dataset: $f : \mathbb{R}^d \mapsto \mathbb{R}$. Our approach is to study the *critical points* of f to have a compact and powerful representation of the signal. Critical points can be of different types:

- local **maximum**: a function f has a local maximum at the point x^* if there exists some $\epsilon > 0$ such that $f(x^*) \geq f(x)$ for all x within distance ϵ from x^*.
- local **minimum**: a function f has a local minimum at the point x^* if there exists some $\epsilon > 0$ such that $f(x^*) \leq f(x)$ for all x within distance ϵ from x^*.



(a)

(b)



(c)

Figure 1.1: Examples of critical points: (a) Minimum and maximum of a 1-dimensional function (b) Saddle point of a 1-dimensional function (c) Saddle point on a surface

- **saddle point**: it is a point of a function f which is a stationary point (i.e. derivative equal to zero) but not an extreme (i.e. maximum or minimum).

Examples of critical points are shown in figure 1.1

The analysis of the scalar field by means of its critical points can dramatically reduce the computation complexity of many problems while maintaining all the relevant information carried by the original signal. Out of the many possible ways to study them, we will focus on two techniques related to the concept of *scale*: Scale-space and Persistence (Chapter 2) analysis.

Chapter 2

Dealing with scale

The concept of scale is inherent in many kind of signals: from one-dimensional signals, to images, to 3D shapes, and even at higher dimensions, every signal can be seen as a composition of structures: a small set of features represents the coarse structure of the signal, while a larger set of details represent finer structures that can be seen only when giving a "*closer look*" at the signal. Since the 80s, many studies have been carried out, with the goal of giving a framework to represent and analyze signals at different scales. In this chapter, we will introduce *scale-space*, describing its most common implementations and their limitations, and we propose an alternative implementation that can overcome such limitations. In the last section, we describe Persistence filtering, a technique coming from topology and Morse Theory, which can be used in the context of multi-scale analysis, either as an alternative, or in combination with scale-space.

2.1 Introduction

Scale-space methods have been greatly studied in the literature, starting from the seminal works of Witkin [Wit83] and Koenderink [Koe84], finding wide usage in computer vision and image processing. See Section 4.4 of [Pri] for an updated bibliography.

Lindeberg [Lin94] introduced the concept of *deep structure* of the scale-space, which captures the evolution of differential properties across scale-space of some differential invariants, by tracking their zero-crossings. Most typically, these invariants are the critical points of a scalar field. A straightforward definition of *importance* is then associated to each critical point as its life time in the scale-space; this measure helps identifying relevant features of the signal.

The mathematical definition of the scale-space, which will be given in the next section, is continuous, but the standard approach is to compute a discrete sampling of it: a discrete scale-space is then made of a collection $(f_0, f_1, \ldots, f_n)$, where f_0 is the input signal and each subsequent

sample is a filtered version of the previous one. Features can be computed at each level f_i, and then tracked across pairs of consecutive levels f_i, f_{i+1} to construct the *deep structure* of the scale-space. The standard approach works as follows: given a feature point p at level f_i, look in its neighborhood at level f_{i+1} for a feature of the same type. If it is not found, then the life of such feature is defined at level i, while it is considered to disappear at the next level. This procedure is prone to error (false and missed matchings) and its greatly influenced by the sampling of the scales: if the sampling is too loose with respect to the rate at which features disappear, there will be more errors.

In [RKG+11], a method was introduced to track critical points based on the gradient of the field to detect correspondences across level; while more robust than the classical approach, is still suffers of the shortcoming related to the discrete approximation of a continuous process. To overcome this drawbacks, in [RP13] an approach was proposed, which computes a virtually continuous model of the scale-space and its deep structure: by employing a piecewise-linear representation of the input signal f and discretizing its domain into a simplicial mesh, the extraction of critical points becomes straightforward. It also adopts a piecewise-linear representation of the diffusion flow that generates the scale-space, by assuming it to be linear between each subsequent pair of levels f, f_i. This makes tracking of critical points easier and precise in the context of the piecewise-linear approximation. Despite its advantages, this method is still a piecewise-linear approximation of a non-linear process. This may still lead to mismatch in the tracking of critical points, when certain particular conditions occur, which make the approximation violate scale-space axioms.

In the following, we study when this situation occurs and we propose a solution to keep the piecewise-linear approximation, while simultaneously keeping the scale-space consistent with the axioms and obtaining a much more robust tracking of critical points.

2.2 Scale-Space Analysis

In this section, we will first present the formal definition of the scale-space, and we will discuss its classical discrete implementation and the consequent problem of tracking critical points across scales. Then, we will discuss the extensions mentioned above, their shortcomings, and our approach to solve them. For the sake of simplicity we will deal with the scale-space of a bivariate scalar function defined on a 2D rectangular domain sampled on a regular grid; most of the concepts are easily generalized to other domains, dimensions and sampling; we will briefly discuss them.

Let $f : \mathbb{R}^2 \mapsto \mathbb{R}$ be a bivariate scalar function. For convenience we will assume that f is a Morse function, i.e. all its critical points are isolated; in applications, where this is not true, we will perturb the function slightly to avoid *plateau* at critical points. The *linear scale-space* $F_f(x, y, t)$ is defined as the solution of the *heat equation*:

$$\frac{\partial}{\partial t} F_f = \alpha \Delta F_f$$

with the initial condition $F_f(x, y, 0) = f(x, y)$ where Δ denotes the Laplace operator (with respect to the space variables x,y), and α a constant that tunes the speed of the diffusion process. F_f can be obtained by either a diffusion process starting at f, or by filtering f with Gaussian kernels whose variance is directly proportional to t.

Let p be a critical point of f, i.e. a maximum, a minimum, or a saddle. Generally speaking, the diffusion process will possibly make p change its position with different values of t, and p will eventually disappear as critical point, by collapsing with another critical point p'. Collapses always occur between two different kinds of critical points: in 2D, a minimum and a saddle, or a maximum and a saddle. In 1D linear scale-space, the number of critical points always decreases, and no new critical point can be generated as a result of the filtering/diffusion process. However, in the 2D case, critical points might also appear at time $t > 0$, an undesirable issue that we will tackle in Section 2.2.1.1.

Tracking p across its life in scale-space provides a trajectory, i.e. a continuous line in the 2D + time domain, which starts at $t = 0$ (or, in case of critical points appearing at $t > 0$, at the moment of birth), and it either ends at the time when p collapses, or it extends to infinity. We call the interval spanned by p's trajectory in the time dimension the *lifespan* of p; this provides a measure of the relevance of the point p across scales.

Scale-space is generally realized via a discrete representation, in which the scale-space is sampled at a finite set of times ($t_0 = 0, t_1, \ldots, t_k$) on a bounded domain D (we will assume it to be a rectangle). A straightforward discrete representation is just a collection of snapshots ($f_0, f_1, \ldots, f_k$), where each snapshot f_i is a sampling of $F_f(\cdot, \cdot, t_i)$ at the nodes of a regular $m \times n$ grid G_D over D.

The work presented in [RP13] introduced a piecewise-linear representation of the domain D, where each sampling point $[r, c]$ is connected to its neighbors to form triangles $([r, c], [r, c + 1], [r + 1, c + 1])$ and $([r, c], [r + 1, c + 1], [r + 1, c])$ and a virtual outer border is added such that it has a value lower than any sample f_i (ideally, $-\infty$). Each discrete snapshot f_i is extended by linear interpolation inside each triangle, thus obtaining a piecewise-linear continuous function defined on the whole domain. With abuse of notation, we will use $f_0, f_1, \ldots, f_k$ to denote both the discrete function and its piecewise-linear version.

Piecewise-linear interpolation is introduced also in the temporal domain: given two consecutive linear approximations of f, f_i and f_{i+1}, we compute the values for f at any time $t \in [t_i, ti + 1]$ and obtain a new snapshot of the scale-space f_t, defined at each point p of D as follows:

$$f_t(p) = \frac{t_{i+1} - t}{t_{i+1} - ti} f_i(p) + \frac{t - t_i}{t_{i+1} - ti} f_{i+1}(p)$$

The problem of tracking critical points as treated in [RP13] is based on the concept of *flip*. Given a and b sampling points in the domain D, we say that the edge (a, b) *flips* at time t if and only if

$f_t(a) = f_t(b)$ while for an arbitrary small $\epsilon > 0$ either $f_{t-\epsilon}(a) < f_{t-\epsilon}(b)$ and $f_{t+\epsilon}(a) > f_{t+\epsilon}(b)$ or vice-versa.

In the temporal discretization, since the approximation assumes a linear process between two consecutive times t_i and t_{i+1}, an edge (a, b) flips in $t \in [t_i, t_{i+1}]$ if and only if $f_{t_i}(a) < f_{t_i}(b)$ and $f_{t_{+1i}}(a) > f_{t_{i+1}}(b)$ or vice-versa. The exact time t can be computed by linear interpolation:

$$t = \frac{(f_{i+1}(a) - f_{i+1}(b))t_i + (f_i(b) - f_i(a))t_{i+1}}{f_i(b) - f_i(a) + f_{i+1}(a) - f_{i+1}(b)}.$$

The main argument of [RP13] is that a point p can change its status from critical to non-critical, or vice-versa, only because of the flip of one edge incident at p; in case a flip affects the state of a point p that was critical, it can either displace the critical point to a neighbor p' of p, or it can destroy a pair of critical points by collapsing the critical point at p with another critical point of different type at a neighbor p' of p.

In order to detect the events that affect critical points and trace their life through the scale-space, all edges that flip between every two consecutive levels are extracted and sorted by time of flip; flips are then followed in order and, starting from the status of critical points at the input signal f, a data structure containing information about each critical point is updated every time one moves to a different position or when two critical points collapse into each other. When reaching the last level, each original critical point of the input signal f can be then precisely tracked trough the whole scale-space until it disappears (or until it reaches its final position, in case it survives the whole smoothing process).

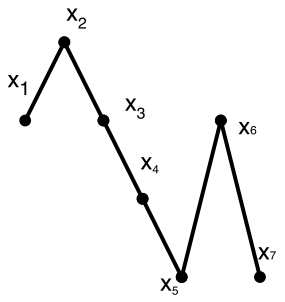
2.2.1 Limitations of the linear approximation

The system presented gives a virtually continuous representation of the scale-space, with the possibility of knowing the state of the system at any time t; under the assumption of linearity between subsequent samples of the scale-space, the tracking of critical points is precise and correct. However, linear approximation of a non-linear process has limitations: in particular, the order of flips obtained by linear interpolation is not necessarily coincident with the true order in the non-linear process. Because of this mismatch, some flips may generate new critical points between samples; this may even occur in 1D, thus violating the *non-creation of local extrema* axiom of the linear scale-space. In other words, the linear process computed between subsequent samples might cause a *flip* to occur at the wrong time, causing the birth of a pair of critical points, which might then survive for a while in the scale-space, causing inconsistencies in the structure of the signal and its analysis.

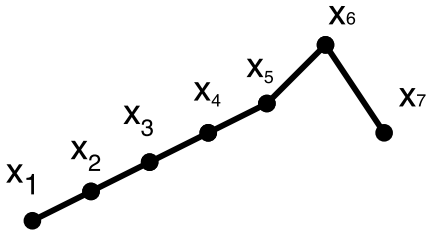
A particular bad situation may occur, in which a pair of critical points vanish, while a new pair of critical points appear nearby a little later. This is often the effect of a wrong order of flips: in the correct non-linear process, the two critical points actually survive and slide through space, while

their sliding is substituted with a death followed by a birth at a displaced location. In this way, unfortunately, we have no way to relate the pair before the death to the newborn pair, thus losing track of the trajectories of such critical points and interrupting their lifespan at a premature stage.

Let us consider, for the sake of simplicity, a signal $f : \mathbb{R} \mapsto \mathbb{R}$ defined by its values at sampling points on a regular grid $\in \mathbb{N}$ and extended to $\mathbb{R}$ by linear interpolation between subsequent samples. An example of such signal is shown in figure 2.1a. For the sake of simplicity, in the figures the values of the points have been exagerated; in fact, intensity of the points in the graph is not relevant, just their relationship, i.e., which one is lower and which is higher.



(a) An example of a signal f defined on a regular grid and extended by linear interpolation



(b) A possible situation for the same signal after the smoothing makes the critical points x_2 and x_5 disappear

Figure 2.1

The smoothing process causes edges to flip, but since the process from 2.1a to 2.1b involves more than one flip (for two critical points to collapse together they must be adjacent, and they have to move via multiple flips to get to that situation), the linear approximation might lead to flips occurring in the wrong order.

The right order in which flips should occur is shown in figure 2.2, but there is no guarantee that that will happen; in fact, the linear process might cause the edge number 2 to flip before edge 1, creating a pair of critical points, as shown in figure 2.3.

2.2.1.1 Avoiding unwanted creations of local extrema

We developed a technique to avoid the creation of new critical points in the linear interpolation of the scale-space process; we process the flips one by one, and each time a new flip is about to happen, we check whether it violates the condition by creating new critical points. If this is the case, we store it in an auxiliary data structure of *delayed* flips and we proceed with the next one. After each successful flip, we scan the list of delayed ones and we check whether it is now possible to process them without creating new critical points; if so, we perform the flip

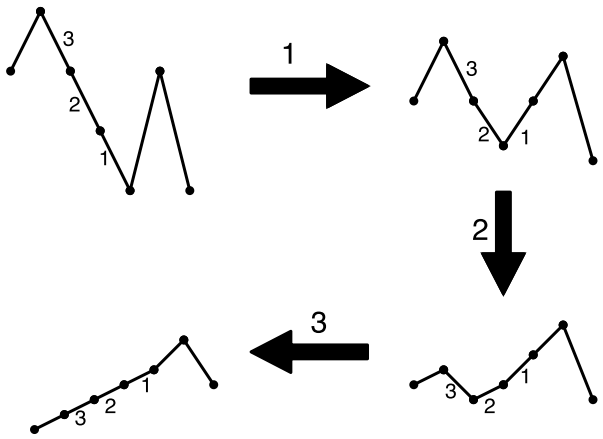


Figure 2.2: The correct sequence of flips that leads to the smoothed signal without creating new critical points

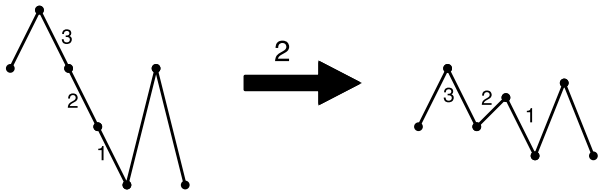


Figure 2.3: In the linear approximation, edge 2 might flip before edge 1, thus creating a new pair of critical points

and remove it from the structure. Figure 2.4 gives a graphical representation of how the process works on a simple example

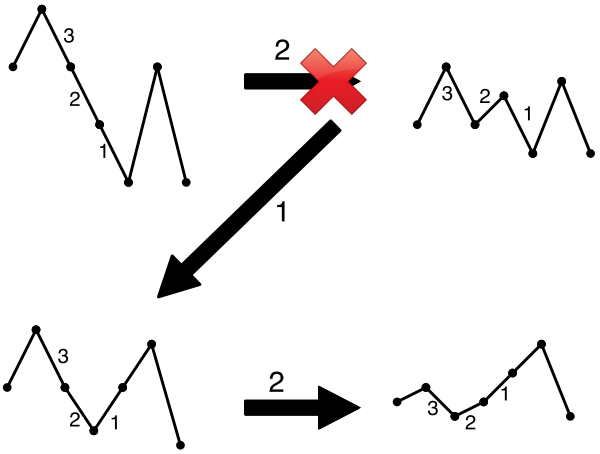


Figure 2.4: The linear approximation would cause flip 2 to occur before flip 1; we delay flip 2 and store it, then we proceed to process the subsequent flip 1; since now flip 2 is legal, we can perform it.

Although this techniques solves most of the cases of flips that would otherwise create critical points, there are some corner cases that are not covered by forcing a delay. Let as consider a situation as the one portrayed in figure 2.5. In this case, from the starting position to the final one we have a pair of maximum-minimum that *slides* to the right with two flips, in the order indicated in the image.

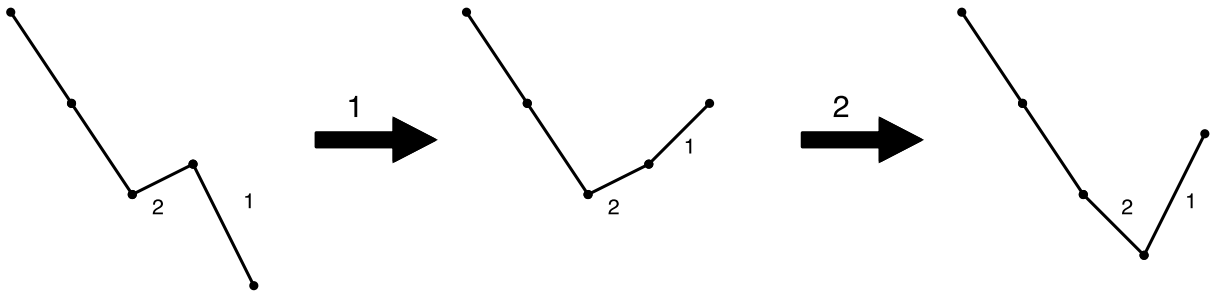


Figure 2.5: A legal sliding of two critical points in two moves: first the flip of edge, which causes a maximum to move on the right; then the flip of edge 2, which causes the minimum to move also to the right,

Now let us assume that the approximation process will cause flip 2 to occur before flip 1, the situation that arises in depicted in figure 2.6. Here the two flips are inverted in order: when flip 2 occurs, instead of having a critical point moving, we have the two critical points collapsing;

since this is a legal move we have no way to know that the flip should have occurred after, so we move to flip 1, that creates an illegal situation by giving birth to a pair of critical points. Since there are no flips after it, we have no way to delay it. It should be noted that the final configuration obtained in these cases is the same, but while this seems make it a lesser problem, it actually carries two issues: we can not detect the wrong order in which we processed the flips by checking the final configuration against the scale-space discrete samples, since we reached the right configuration, and we now have two critical points that we mistakenly marked as dead while they should actually be recognized as the ones we created, to keep tracking them in the scale-space.

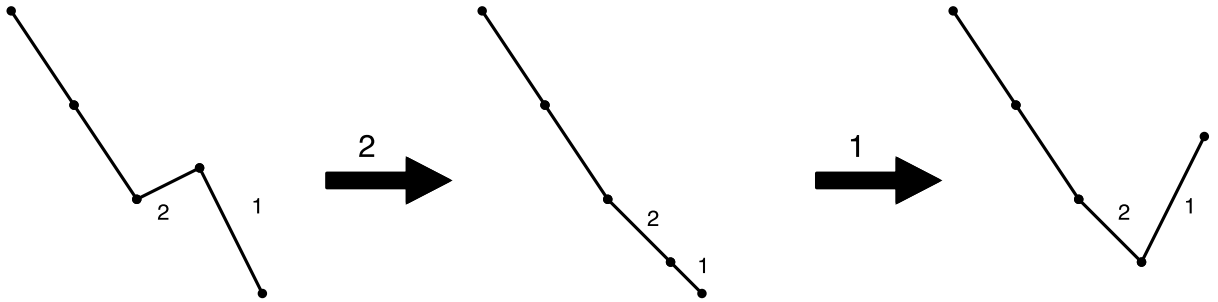


Figure 2.6: An illegal situation arises when the order of the flips is wrong: flip 2 occur first, but since is a legal move (collapse of two critical points) we let it occur, then when flip 1 is about to get processed, we should avoid it since it creates new critical points, but we can not, since it can not be further delayed, because it is the last flip occurring

We provide a solution to this problem too, which makes use of another auxiliary data structure that stores information about every flip that occurs; it keeps in memory what kind of movement it involved (either the death of two critical points collapsing together, or the slide of a critical point). This information is then used when we face an illegal flip which can not be delayed further and has to be executed. The process can be summarized as follows:

1. For every flip f between levels i and $i+1$ of the discrete sampling of the scale-space, ordered by time:
   (a) For every flip f_q in the *delayed queue*:
      i. If f_q can be executed without creating critical points, execute it, remove it from the queue and go back to 1a
   (b) If flip f can be executed, process it and update the scale-space structure.
   (c) Otherwise add it to the *delayed queue*
2. For every flip f_q in the *delayed queue*:

(a) If f_q can be executed without creating critical points, execute it, remove it from the queue and go back to 2

3. For every flip f_q remaining in the delayed queue:

   (a) Create two new critical points M and m (respectively the maximum and the minimum created by the flip)

   (b) Look amongst all the flip occurred in the current slice (i.e. the interval between two discrete samples). Locate the two nearest (spatially) flips, who caused the death of two critical points, occurred before f_q. Let them be f_l and f_r. Get the one between f_l and f_r that is closest in time to f_q. Mark the two critical points collapsed during that flip as alive and update their trajectory with the current position of M and m.

   (c) Go back to 2

Note that points 1a and 2, each in the respective loop, are executed after each flip is processed (either a *regular* flip or a *delayed* one), since delayed flips can become legal after each modification of the structure.

2.2.2 Conclusions

The developed technique described in the previous section allows a representation of the scale-space which is not only virtually continuous (giving us a better control on the tracking of critical points), but also in which the non-creation of extrema is guaranteed. It should be noted that tracking of critical points becomes *exact*, in the sense that each critical point of the whole scale-space is precisely tracked from the input signal, without spurious extrema. Each critical point is now *exactly* represented by its initial position, its trajectory through scales, and the moment in which it gets smoothed out (its *lifespan*), giving us a robust and reliable measure of importance for critical points of the signal.

2.3 Persistence

While scale-space is the most famous and well-studied technique to study a signal at different scales, there are different approaches that can capture the relevant information that subsume the structure of the signal. In this section we will focus on the concept of *persistence*-based filtering.

While scale-space techniques compute a filtering by *frequency* (the Gaussian filter is analogue to a low-pass filter, which blocks high frequencies, i.e. details), persistence can be seen as a filtering by *amplitude*, which relies on the relative values of the samples of the function f.

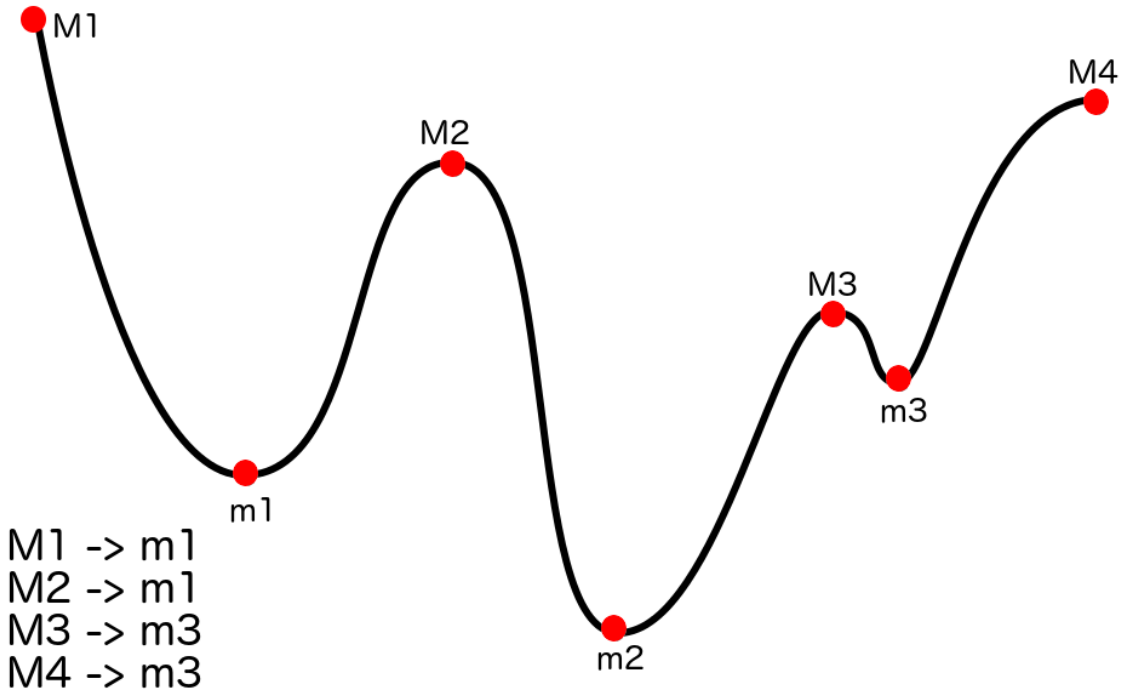
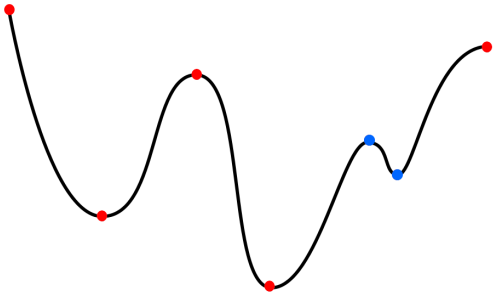


Figure 2.7: An input signal f with its critical points highlighted in red and an initial pairing.

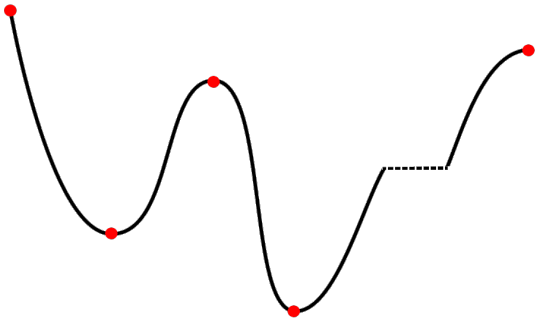
Persistence is a concept related to the Morse theory [ELZ00], which provides a characterization of a function in terms of topology of its level sets. In the following, we will provide a constructive definition of the persistence of a one-dimensional signal, although the general concept holds and is applicable to other signals as well.

Intuitively, persistence in a one-dimensional signal is obtained through a flooding process of *basins,* each expanding from a local minimum: each time a basin gets filled at one of its sides (i.e., along the path connecting its local minimum and the lowest of its two adjacent maxima), this basin gets merged with one of its two neighbours; contextually, the minimum corresponding to the basin and the maximum that has been flooded are removed, and the persistence value of both of them is set at their difference in amplitude. Notice that the merge of basins and the corresponding deletion of flooded pairs of critical points changes the adjacencies of basins: a minimum and a maximum that were relatively far in the original sequence become adjacent during the process, when all critical points between them have been filtered out. For a graphical depiction of the process, see Figure 2.8.

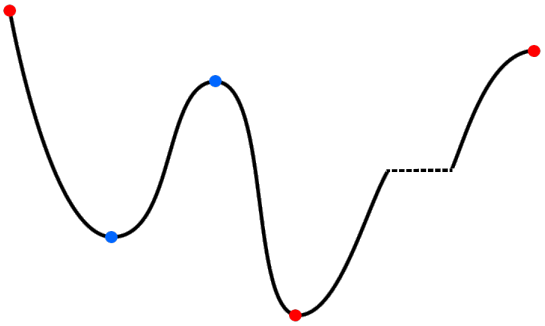
More formally, let us consider a piecewise-linear function f defined discretely by its value at



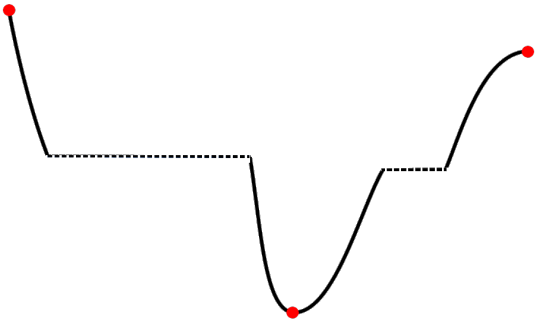
(a) The first step of the filtering: a pair of critical points that is about to be removed is highlighted in blue.



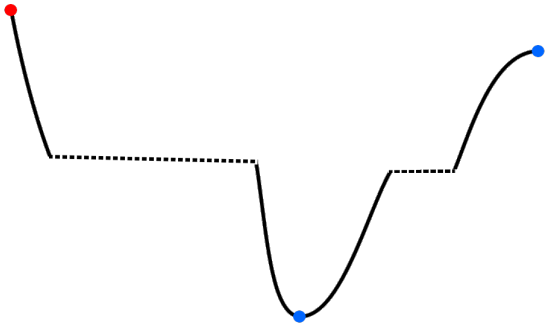
(b) An equivalent signal after removing the critical points: the basin located at the removed minimum has been merged into the one to its left.



(c) Another pair of critical points to be removed is marked in blue.



(d) An equivalent signal after removing the critical points: just one basin has been left.



(e) The last pair of critical points to be removed is marked in blue.

Figure 2.8: Steps of persistence computation. Dotted lines are just placeholders to denote that in such interval the function is considered *as if* it were monotonic. Notice that the function is *not* modified by the algorithm, only the relative adjacency of critical points changes.

sample points regularly spaced in the integer domain and extended to $\mathbb{R}$ by linear interpolation, and the set of its critical points. A point i is called

- a *maximum* (M) if $f(i) > f(i-1)$ and $f(i) > f(i+1)$
- a *minimum* (m) if $f(i) < f(i-1)$ and $f(i) < f(i+1)$

The algorithm computes persistence as follows:

1. Initially, each maximum M_i is paired to its neighbouring minimum m_j, whose difference in value from M_i is minimal. Figure 2.8 (a) shows an example of the initial pairing. We take a bookkeeping of the current pairings in a map data structure, as they may become obsolete during processing (see point (3), last step).
2. Then, all maxima are stored in a priority queue Q, where priority is set by the difference in value between M_i and the minimum paired with it: maxima with lower difference have higher priority.
3. Maxima are progressively popped from queue Q. Each time a maximum M_i is popped: if its pairing has become obsolete, then it is discarded; otherwise, the absolute difference between M_i and its paired minimum m_j is written in output as their persistence, and the pairings of their neighbouring critical points are updated as follows:
   - M_i and m_j are excluded and marked as *inactive*; from a theoretical point of view, this is like removing the pair of critical points selected (e.g. the two blue points in figures 2.8 (b), (d), and (f)), thus obtaining a new function with two less extrema (figure 2.8 (c), (e)). From the perspective of the flooding process, it means that the basin with the local minimum in m_j is merged into the adjacent one on the other side of M_i.
   - The pairing of the closest active maximum M_k on the other side of M_i with respect to m_j might need to change (in case it were also paired with m_i): in this case, the new pairing of M_k is determined by looking at its active neighbuor minima and choosing the one with the closest value to it. The bookkeeping of pairings is updated by assigning to M_k a new paired minimum, and M_k is then reinserted into Q, thus making any older instance of itself in the queue obsolete; since the auxiliary data structure is always up-to-date, an instance of a critical point M_k in Q is considered obsolete if and only if its paired minimum m is different from the one in the bookkeeping.

   The process ends when there are no more elements in the priority queue. The last maximum remaining in the filtered signal is assigned an arbitrarily high value of persistence, larger than the maximum persistence computed previously.

At the end of the process, each critical point c_k of the input signal has been assigned a real number p_k, which is the computed persistence for that critical point.

2.3.1 Conclusions

The introduction of the persistence analysis as another approach to multi-scale analysis introduces a different concept of *filtering*: while the scale-space generates different version of the same signal, persistence filtering does not modify the input signal. Moreover, the scale-space is tuned by a couple of parameters (the number of levels generated, and the size at which the Gaussian kernel grows between each one); changing them implies a different final representation (although the linear interpolation between levels makes the impact of the parameters less drastic). Persistence analysis, on the other hand, does not depend on any parameter, but only on the values of the critical points of the input signal and their relationship of neighborhood.

However persistence filtering gives, analogously to the scale-space analysis, a ranking of importance of critical points of the input signal, in which the importance of every critical point is now its persistence value. The two rankings obtained represent different properties of the signal analyzed, but they can be compared and combined. This is discussed in the next section.

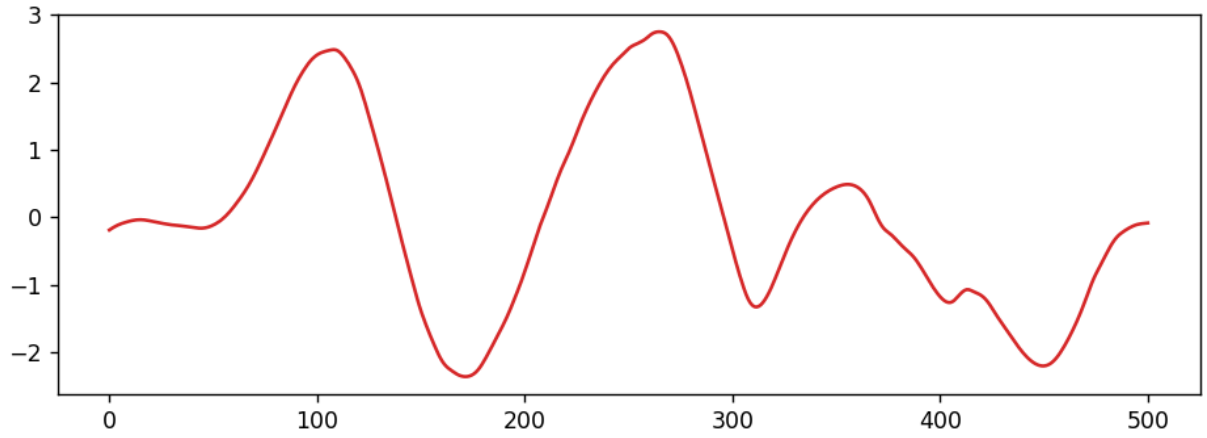
2.4 Combining the multi-scale analysis

Both the scale-space and the persistence analysis focus on the critical points of an input signal f and aim at extracting information related to the structure of the signal at various *scales*. However, being two process that work in different domains (frequency and amplitude), the results obtained by the two analyses might differ.

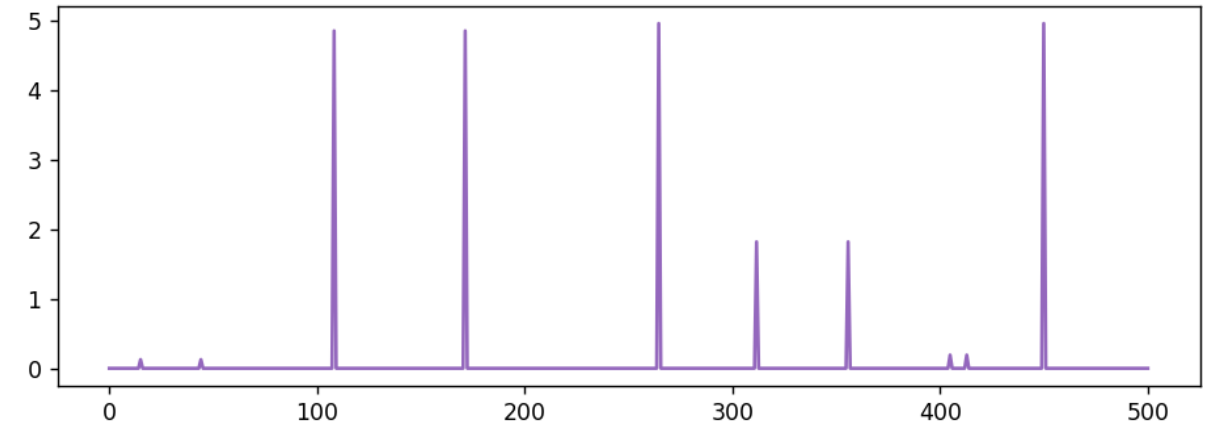
Figure 2.9 shows an example of an input signal on which persistence and scale-space analysis have been carried out. While there are obvious similarities between the two results, it can be seen that pairing between maxima and minima are different, as well as relative intensities of the values. However, the two analyses can be carried out independently and then combined together to obtain a more powerful and complete representation: consider that both processes associate to each critical point c_k of the input signal a real number, either its life in the scale-space, or its persistence. More precisely, in the case of the scale-space, each critical point c_k of the input signal is associated to a real number l_k, representing an approximation of how much it will survive a continuous smoothing process, and the result can be converted into a signal S which is the composition of impulse signals, in which impulses are located where the original signal had critical points, and whose height is the lifespan of such critical points. In the 1-dimensional case, it can be expressed as follows:

$$S(i) = \begin{cases} l_k & \text{if } f(i) \text{ is a critical point } c_k \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

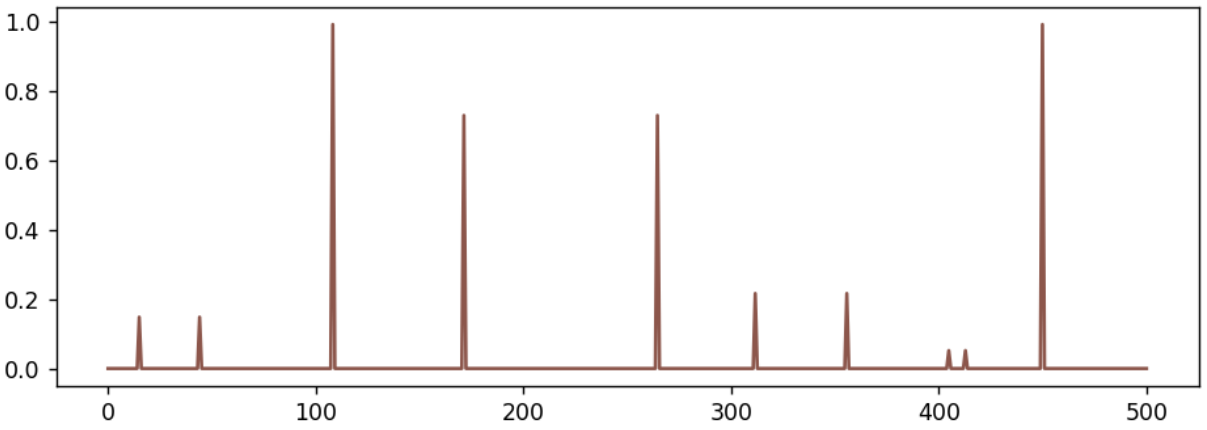
while, in the case of the persistence, each critical point c_k of the input signal is associated to a



(a)



(b)



(c)

Figure 2.9: A 1D signal (a) and its corresponding impulse signals for persistence (b) and lifespan in the scale-space (c). Note that even critical points which are far in the signal can be smoothed out together (those with the same value of persistence).

real number p_k, which is the computed persistence for that critical point.

As in the previous case, we can encode the result of our analysis into a signal P composed of impulse signals, which in the 1-dimensional case will be defined as follows:

$$P(i) = \begin{cases} p_k & \text{if } f(i) \text{ is a critical point } c_k \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Note that S and P are congruent impulse signals that have non-null values only at critical points of the input signal f. The two signals might have very different magnitude in the impulses: those of S, i.e. the lifespan of critical points in the scale-space, depend on the frequencies of the input signal, the size at which the Gaussian kernel grows, and the total number of levels computed; the impulses of P, on the other hand, depend on the differences in amplitude between critical points, and it can never exceed the difference between the highest and the lowest point of the input signal.

In order to combine the two signals, we normalize them:

$$P_{norm}(i) = P(i)/max_P \tag{2.3}$$
$$S_{norm}(i) = S(i)/max_S \tag{2.4}$$

where max_P and max_S are, respectively, the maximum values of P and S.

At this point both signals have values in the $[0, 1]$ range, and we can combine them into what we called the Multi-Scale Index (MSI for short), as follows:

$$MSI(i) = (\alpha \cdot P_{norm}(i)) + ((1 - \alpha) \cdot S_{norm}(i)) \tag{2.5}$$

where $\alpha \in [0, 1]$ is a parameter that weights the contribution of each signal.

!TeX root = ../main.tex

Chapter 3

Motion Capture Data

In this chapter we will introduce Motion Capture, a technology that has seen a continuous rise in popularity during the last decades, becoming more and more relevant in many computer science fields, such as bioengineering, human-computer interaction and motion analysis. We will discuss its many applications and the different acquisition techniques developed in the past, and we will focus on passive markers acquisition, discussing the challenges that its analysis poses.

3.1 Introduction

Motion capture (Mo-cap or Mocap for short) is the process of recording the movements of objects or people; it has many applications, such as animation, rehabilitation, robotics, and many more. The subject recorded is usually a person during some kind of activity, which may range from simply walking and doing basic tasks to more complex activities like dancing, playing an instrument, or doing sports. For instance, in medical applications this can be used to analyze issues in basic movements. Possible extensions include recording the activity of a group of independent subjects acting at the same time; this last scenario creates a more complex context that will be discussed later.

The motion is usually sampled at a very high frequency, in the order of hundreds times per second, making it possible to capture complex movements with a relatively small effort (with low-cost setups, some postprocessing of the data might be necessary, though); on the other hand, specific hardware and software are required, and it is not always easy to switch from a system to another. Furthermore, the particular hardware used poses limitation on the space it can be used, for example due to the camera's field of view. Lastly, Mocap systems are usually very expensive. However, in the last few years there has been an explosion in the production and marketing of cheaper solutions (e.g., Microsoft Kinect), although with limitations with respect to the high-end

systems (e.g., Vicon, Qualisys).

3.2 Applications

Motion capture has been used often in the video games industry to animate athletes, martial artists and other in-game characters. By 1995 motion capture was a common technique used in video game development, at the point that the developer/publisher Acclaim Entertainment, one of the leading developer of martial arts games, built its own in-house motion capture studio in its headquarters.

Gait analysis in another important field in which motion capture has seen an increasing use in the last decades; it is the major application of motion capture in the clinical medicine, in which clinicians are able to evaluate human notion across several biometric factors.

One of the main applications in which motion capture became a worldwide success is, however, the creation of Computer Graphics (CG) effects for movies; most movies containing computer-generated creatures used motion capture to track human actors who will be later transformed into their characters, including Academy Awards Visual Effects winner such as *Avatar*, *King Kong*, *Pirates Of The Caribbean*. Nowadays, most of the movies (both animated and not) involving computer-generated creatures make use, up to some degree, of motion capture systems.

3.2.1 Animation database

An animation database stores fragments of animations or human movements, providing tools to access, analyze and query them in order to develop and assemble new animations [Sum09, KKKM93]; they are systems designed to help a user to build animations starting from existing components, in order to dramatically reduce the time and effort needed to generate a large amount of animation. A concrete issue in building an animation database is the source of the fragments: one way to create them would be to separately record small sections of movements and using them as basic movements; however, this would be really time consuming and would not be retro-compatible with prerecorded motions. Another approach might be to manually segment the movement, but that will not just be biased by the subjectivity of the process, but it will also require a lot of time.

A possible solution, which will solve both the time and the retro-compatibility issues, would be to automatically segment longer recordings of movements into a set of basic ones; we will discuss this possibility in section 5.

3.3 Acquisition Process

A number of different techniques have been developed for motion capture, with a variety of different properties, features and limitation; we will shortly describe the most widely used, and we will focus on the one used for the data we worked with.

3.3.1 Non-Optical System

This class of method contains those that do not extrapolate 3D data from optical images

- **Inertial systems**: is a technology based on inertial sensors placed on a person, that wirelessly transmit their motion data to an external receiver (e.g. a computer); most systems use inertial measurement units (IMUs), containing a combination of gyroscope, magnetometer and accelerometer, to measure rotations that are then translated to a skeleton in the software. The more IMU sensors are placed on the person, the more natural the movement that can be reproduced from the data. Moreover IMUs alone can not give information about the absolute position of the user.
- **Mechanical motion**: they directly track body joint angles; they are usually referred to as exoskeleton motion capture systems, due to how the sensors are attached to the body. A performer gets a skeletal-like structure attached to his body and then moves the mechanical parts as if they were her joints.
- **Magnetic systems**: they work by emitting magnetic flux from both the transmitter and each receiver; the relative intensity of the voltage of three orthogonal coils allows these system to calculate both range and orientation; these systems are not occluded by non-metallic objects but are obviously subject to magnetic and electrical interference.

3.3.2 Optical System

Optical systems are those that utilize data from image sensors to triangulate the 3D position of a subject seen from two or more cameras. Some of them are described below:

- **Passive markers**: these systems make use of markers coated with a retro-reflective material to reflect light that is generated near the camera lens. The system is calibrated by moving an object with markers attached at known positions; with this system it is also possible to measure the distortion of each camera. When an object with markers is placed on the scene, if two calibrated cameras can see a marker, its three-dimensional position can be obtained; usually a system is made up of around 10-20 cameras, in order to reduce

the risk of marker swapping (each marker looks identical to the cameras) and occlusions. Cameras have also usually a very small field of view, so many of them are required if the need to track objects in a large space exists.

Unlike other methods, passive markers systems do not require the user to wear wires or other electronic equipment thus giving more movement freedom; the tracking is realized by velcroing small rubber balls with reflective tape on specials full body suites. If the tubber balls can not be attached to an area that must be tracked (e.g. fingers), usually small velcro strips covered by a reflective tape are placed instead. An example of a passive marker setup is shown in Figure 3.2.

Passive markers systems can capture a large number of markers at high frame rates: usually they can go quite easily to 250fps, but by lowering the resolution and tracking a smaller region of interest they can go as high as 10000 fps (in the most modern and expensive systems).

This kind of system is used to acquire the data for the analyses presented in this thesis.

- **Active markers**: they triangulate positions by illuminating one LED at a time very quickly or multiple LEDs with software to identify them by their relative position; rather than reflecting external lights as in passive markers, in this case the light comes directly from the markers. This increases distances and volume for capture, as well as high signal-to-noise ratio and resolution.

- **Markerless systems**: it is an emerging research field, based on computer vision techniques. It does not require the user to wear special equipment, instead it rather relies on algorithms to analyze multiple streams of optical input and to identify human forms, breaking them down into constituent pars for tracking.

  The most popular example of markerless mocap is the one performed by Microsoft's Kinect. The 3D information is extracted using a patented depth sensor which combines a classical camera and an infrared projectors which paints the scene with invisible markers that can be seen from an infrared camera. This gives the system a way to triangulate the position of each point in the scene: an infrared speckle pattern is projected into the scene; whenever the camera sees a point in the scene, the system recognise the pattern and knows the trajectory from that point to the projector, and thus it can triangulate the position of the point. The association of the depth information with a human skeleton, and thus the tracking, is done with a computer vision algorithm based on the RGB information. A graphical depiction of the process is shown in figure 3.1. A major downside to this system is that multiple kinects will interfere one with another, as infrared projection will overlap, so it is not possible to have 360°tracking of the person in the scene.

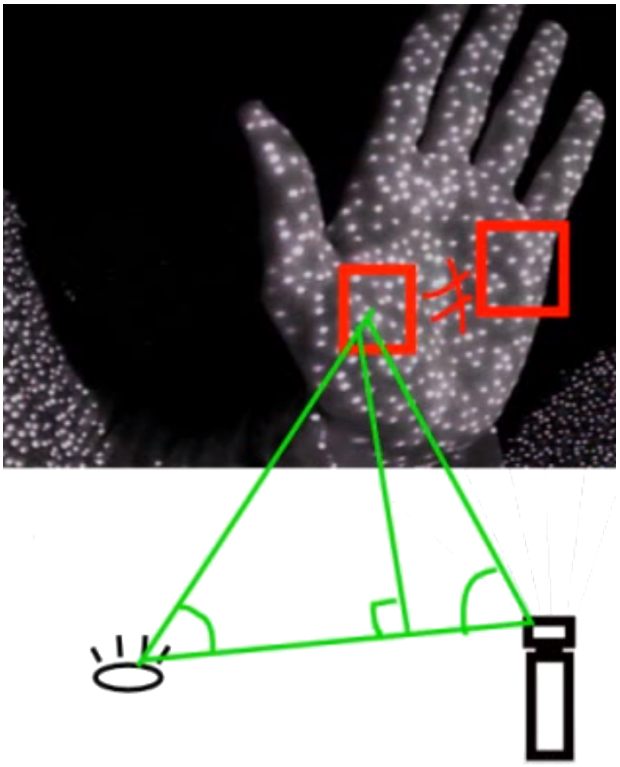


Figure 3.1: An example of how the kinect depth sensors works. In the upper part we can see the infrared speckle pattern. The system is able to differentiate between each possible pattern; it then knows the relative position of the camera and the projector, as well as the angles marked in green; it can then triangulate the position of the point onto which the pattern is projected.

Figure 3.2: The setup of passive markers and reflective strips (they can be seen on the hand and feet) for a Mocap session

3.4 Data

For all the experiments in our thesis, we will consider data coming from a *passive markers* system. More precisely, the data were recorded using a *Qualysis* system, composed of 9 high-resolution cameras and the related proprietary software. An example of setup of markers can be seen in Figure 3.2; although not all recordings were realized with this specific setup, the general idea of markers attached to the body, or to a spandex full-body suit, is always present in passive markers systems. The particular setup also includes a microphone and IMUs (attached with the large white strips on ankle and forearm), but it has not been taken into account in our work.

Data coming from these sensors are processed on-line by the cameras themselves, which sample the position of the markers at the frequency they are set to. For each marker, a triple (x, y, z) representing the triangulated position of that marker at each instant in the calibrated reference system is obtained.

Given a setting in which n markers are tracked, with a recordings of f frames, we can define our data as:

$$C_{ij} = (x_{ij}, y_{ij}, z_{ij}) \tag{3.1}$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, f$ and C_{ij} is the position of marker C_i at instant j.

3.5 Analysis

Data coming from mocap systems pose many challenges, since there are many factors involved in the generation, collection and processing of the data that influences its size and nature:

- Human motion is in itself very complex and can greatly vary: the same movement can be done with different features, as widely studied in the literature, from the seminal works of Laban [vLL74]; furthermore there can also be variety at lower level features, since even the same person doing the same movement twice with the same intent will not precisely move through the same space and at equal speed.
- The quality of data gathered depends greatly from the set-up of the system: to maximize quality, markers should be placed in strategic positions on the human body, and great care should be taken to avoid markers sliding from their placement or even falling down from the suit. Furthermore, even when many cameras are used to track the body, there is always the chance of occlusions and mismatch of markers, since every marker looks the same to the camera.

- A high frequency of sampling (e.g. 250hz) on a relatively large set of markers (we worked with settings up to 64 markers), means that every second we record $64 \cdot 250 = 16000$ triple consisting of (X, Y, Z) double values. This means that a relatively small recordings of a couple of minutes consists of almost 2 millions triples that need to be processed in an adequate way to reduce computational complexity and extract relevant features to reduce the cardinality of the problem.

All these considerations lead to the need of suitable algorithms and techniques to deal with noise, dataset size and variability of the provided data; in the next chapter we will employ multi-scale techniques to extract and represent features useful for analysis.

Chapter 4

Analysis Of Synchronization from Karate's motion capture recordings

In this chapter, we present a method to measure intra-personal synchronisation of movement from motion capture data, and we show that our method is effective in classifying the level of skills of athletes performing karate kata. Our method is based on detecting relevant peaks of acceleration of limbs (arms and legs) and measuring their synchronisation. We run a multi-scale analysis, based on topological persistence, to rank the importance of peaks of acceleration. The resulting impulse signals are processed next with a Multi-Event Class Synchronisation algorithm, and we define an *Overall synchronisation index* that scores the level of intra-personal synchronisation with a single scalar value. We build a basic multiclass classifier, which uses just the means of indexes computed on the different classes in the training set. We make a statistical analysis and a cross validation of the classifier on real data. Performances by athletes from three levels of skill have been recorded, classified by experts and used to test our method. Cross validation of the classifier is performed by leave-one-out and bootstrap resampling. Results show that our method can classify correctly with very high probability (beyond 99%), while it succeeds on 100% of the data used in cross validation.

References:

Paolo Alborno, Nikolas De Giorgis, Antonio Camurri, and Enrico Puppo. 2017. Limbs synchronisation as a measure of movement quality in karate. In Proceedings of the 4th International Conference on Movement Computing (MOCO 2017).

Nikolas De Giorgis, Enrico Puppo, Paolo Alborno and Antonio Camurri. 2018. Evaluating movement quality through intra-personal synchronisation. Submitted for review to Behaviour And Information Technology Journal.

4.1 Introduction

The automated analysis of human full body movement to investigate movement qualities has been recently investigated by a wide number of studies; it is a challenging task due to the many issues that arise from what is described in section 3.5.

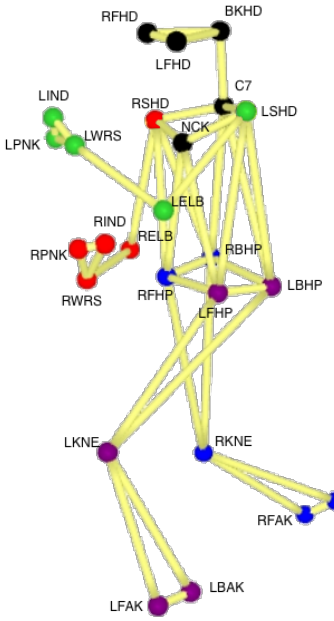
Features describing quality of movement can range from physical low level measures (velocity or acceleration of certain parts of the body), to more complex derived ones, such as coordination, balance and synchronization between the different parts of the body.

Our goal is to find an automated way to evaluate the overall quality of martial-arts performances from Motion Capture recordings, focusing on the case of karate. More specifically, we try to quantify the quality of a karate's performance, at different levels of expertise, by means of the level of synchronization between the movements of the limbs. The idea behind studying synchronization as a measure of quality is rooted on well known common assumptions: an experienced athlete makes a more neat execution, with starting and ending phases clearly identified; movements exhibit less fluctuations or ripples with respect to a less skilled athlete, and they are strongly synchronized between the limbs executing them (i.e., the arms or the legs). This also corresponds to the concept of *soft entrainment* used in other contexts, such as music performance [YTY02]. Stability is obtained by a high level control of the movement, emerging from a strong synchrony between limbs; this is particularly true in the execution of the basic elements of a performance, such as punches, strikes and kicks.

We consider a set of performances (*katas*), which are analyzed aiming at distinguishing and classifying each performance on a measure of the overall quality. As previously discussed in 3.5 there are several potential issues while analyzing this kind of data: although some processing aimed at cleaning the raw data has been done, performing sessions are of different length, due both to the different speed at which the athletes perform and to the fact that recordings do not start and stop exactly at the beginning and end of the kata; motion capture data can be very noisy, not just because of the acquisition process, but also because of intrinsic noise in the biological movement itself. These issues are tackled by using a multi-scale analysis to get rid of the noise and extract relevant information present in the input signal, which is then fed to an event synchronization algorithm.

The data we used has been presented in [KCV+15]; the recordings have been realized using an optical system with passive markers (see section 3.3.2 for a more accurate description), using 9 high resolution cameras set at a 250Hz frame rate. Some post-processing on the data has been applied in order to reduce the noise from *ghost* or jitter marks.

A total of 7 athletes participated in the recordings; they were chosen with different skills and levels which represent a variety of abilities of execution; their ability has been assessed by experts on a conventional scale from 1 to 5, while only levels from 3 to 5 are represented in the dataset. The participants performed two different katas, namely *Heian Yondan* and *Bassai Dai*. Each



Left Arm	LIND	Right Arm	RIND
	LPNK		RPNK
	LELB		RELB
	LWRS		RWRS
	LSHD		RSHD
Left Leg	LFHP	Right Leg	RFHP
	LBHP		RBHP
	LKNE		RKNE
	LBAK		RBAK
	LFAK		RFAK

Figure 4.1: The MoCap skeleton. In red, green, purple and blue the groups of markers used to define the four clusters, summarised in the side table.

athlete performed 2 or 3 trials of each kata. The final datasets consists of 32 recordings, divided as follows:

- *Level 3 - Junior, Brown belt*: 7 recordings of the Heian Yondan kata and 7 of the Bassai Dai
- *Level 4 - Senior, Black belt 1st dan*: 4 for Heian Yondan and 5 for Bassai Dai
- *Level 5 - Master, beyond 4th dan*: 5 for Heian Yondan and 4 for Bassai Dai

Athletes were let free to perform the kata at their own speed and rhythm with no interventions, even in post processing, to match the lengths of the various trials or align the recordings. As a result, recordings have various differences both in length (*Bassai Dai* ranges from 71 to 115 seconds, *Heian Yondan* from 50 to 97 seconds), and also in the relative speed of parts of the kata.

Motion tracking has been realised with 25 makers placed on the body (as shown in Figure 4.1), each providing a 3D trajectory of samples at 250hz, where every sample consists of a triple (x, y, z) of coordinates representing the position of that marker at each time instant, in a calibrated reference system.

Since our analysis is focused on the movements of the limbs, we do not use every marker of the model but we rather extract four clusters (one for each arm and each leg) in order to exploit

redundant information and to obtain a reduced yet more stable representation. See Figure 4.1 for a definition of clusters.

4.2 Related Work

Modeling and analyzing qualitative bodily expressions requires an interdisciplinary approach. Contributions from experimental psychology and neuroscience ([MF69]; [Arg88]; [DM89]; [Wal98]; [BC98]; [PHM03]; [DG09]), robotics and human movement sciences ([KVY+16]), artistic and humanistic theories and in particular dance and choreography (e.g., Rudolf Laban's Theory of Effort [LL47]). Several systems and multimodal interfaces for the automated analysis of affective bodily states have been proposed: a survey and review is available in [KBB13]. Several existing systems start from video signals and are grounded on computer vision techniques ([CLV03]; [KKVB+05]; [GP09]; [BR07]). Real world applications have been proposed in various fields, including education, games, social inclusion, therapy and rehabilitation, in controlled settings. For example, in [KKVB+05] sadness, joy, anger and fear are detected using acted full body motion captured data. Gunes et al. (in [GP09]) proposes a vision based bimodal system which tracks face and upper body motion included fear and anxiety among twelve affective states. From the perspective of continuous affective dimensions, Kleinsmith and Berthouze ([KBBS11]) applied MLP to predict valence, arousal, potency and avoidance with recognition rates comparable to human observers. Emotional dimensions are grounded on movement qualities, and some of them are relavant for the evaluation of sport movements, such as anxiety, hesitation, fluidity and coordination.

Several methods have been proposed in the literature to evaluate the overall quality of martial arts performances. A variety of techniques have been employed: Bianco and Tisato ([BT13]) proposed an algorithm for karate movement recognition from skeletal motion on a dataset consisting of punches, kicks, and defense karate moves. More recently,Kolykhalova et al. ([KCV+15]) presented a multimodal dataset of karate performances including synchronized MoCap, video, and audio recordings and considered a set of global measures, which can be used to evaluate the quality of the performances. Several studies have focused on investigating the movement of specific body parts: Vieten and Riehle in [VR08] focused on the quality of kicks in martial arts, while [VFC+11] present a kinematic and electromyographic analysis of punches during the performance of a particular kata (choku-zuki).

4.3 Extraction of events

From the clusters shown in figure 4.1 (each one represented by its barycenter) we want to extract relevant instants and to compute their synchronization; the feature that we found to be more

representative for our analysis is given by peaks of limbs' acceleration (and deceleration) that allow us to distinguish the initial and final phases of the basic movements, such as punches, strikes, kicks, steps, parry and block actions. Since recordings are quite noisy, in order to perform a stabler analysis, we extract a smoothed version of this feature: we compute the velocity of the clusters' barycenters at each frame, then we perform a moving average filtering of this signal, going from the raw velocity signal v to the smoothed sv as follows:

$$sv(i) = \frac{1}{l} \sum_{k=-\frac{l-1}{2}}^{\frac{l-1}{2}} v(i+k)$$

where l is an odd number representing the width of the filtering window. Acceleration is then extracted by computing sv's derivative. We found that a value $l = 41$, corresponding to slightly less than a sixth of a second, provided a reasonably smoothed signal without sacrificing relevant details in all our experiments.

Peaks of acceleration and deceleration characterise instants of time that are relevant for our analysis, but they also appear along trajectories, because of noise, uncertainty and ripple in the movement. Since data are rather noisy, isolating relevant peaks from unimportant ones is a challenging task, which we tackle by computing the topological persistence (see 2.3) of the extracted acceleration.

4.4 Synchronization

The output of persistence analysis consists of the following time-series:

$$ts_i \quad \text{with} \quad i \in \{leftArm,\ rightArm,\ leftLeg,\ rightLeg\}$$

containing the values of the Persistence Index (PI) at each frame.
We now consider pairs of time series ts_i and ts_j, related to different limbs, and we measure their mutual synchronisation, by finding matching events and measuring their difference in time. Our method relies on two parameters:

1. A threshold ρ for the *PI* of events to be considered relevant;
2. A maximum time lag τ between synchronous events.

We first threshold both time series, saving just relevant points with values $PI > \rho$. Then, we apply an extended variant of the Event Synchronization Quian Quiroga's algorithm [QKG02]: the Multi Event Class Synchronization (MECS) algorithm.

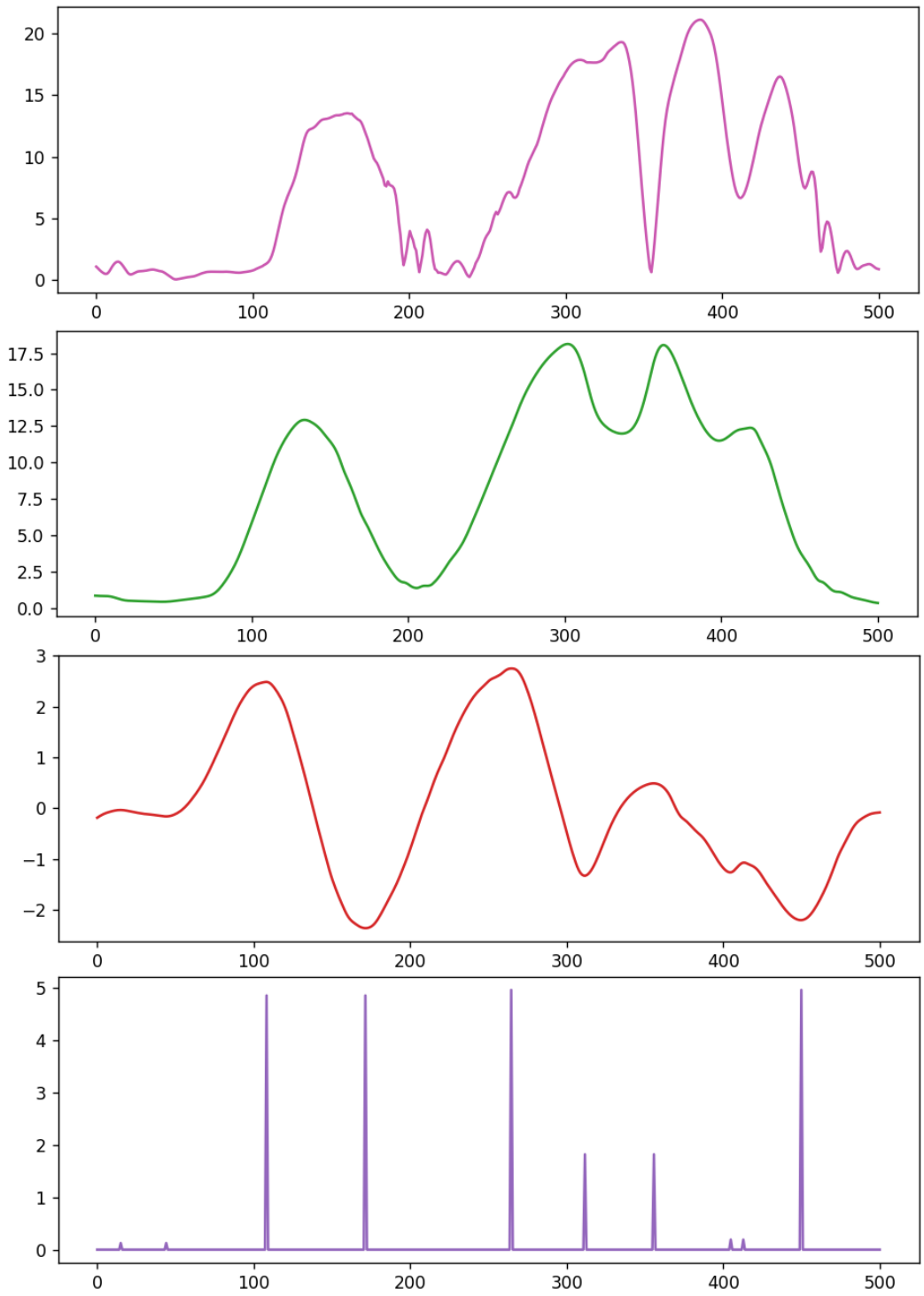


Figure 4.2: From top to bottom: the original velocity extracted from the clusters' barycenter; the smoothed version with a moving average; its derivative, i.e. our acceleration; the persistence values of the acceleration

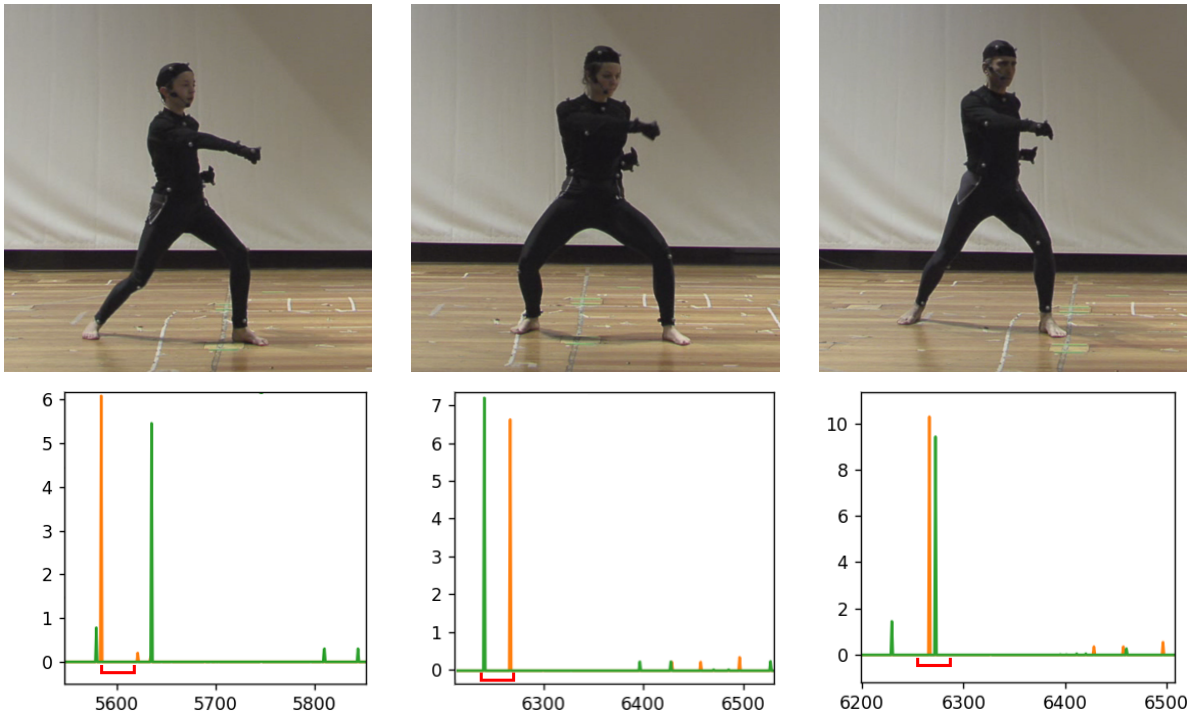


Figure 4.3: The same movement, the ending of a punching phase, performed by athletes at different levels (from left to right: level 3, 4 and 5); and the corresponding persistence values: in orange persistence values of the right arm, in green those of the left arm. Movements at higher levels are highly synchronized while the synchronization decreases with the level. The red bracket is the size of the synchronization window $\tau = 40$ used for our analysis, showing that movements of the Level 3 performer will not count as synced, while both Level 4 and Level 5 will contribute positively to the final synchronization index, but with different intensities

Similarly to [QKG02], MECS consists of two steps: the algorithm first detects the coincident events in different time series (*coincidence detection*) and counts them; then the number of coincidences is normalized with respect to the total number of possible coincidences that may happen (*normalization*). Besides, MECS modulates the contribution of each pair of coincident events as a decreasing function of their difference in time.

Let t_i^x with $x = 1, ..., m_i$ and t_j^y with $y = 1, ..., m_j$ be the times of events x and y occurring in sequences ts_i and ts_j, respectively, where m_i and m_j represent the total number of events in the time-series ts_i and ts_j, respectively. A pair of events x and y contribute to the synchronisation index if and only if they occur within a time interval (coincidence window) not larger than τ.

Coincidences are detected by a simple parallel scan of the two time series, like in a merge algorithm for sorted sequences: pairs of events x and y that are consecutive in the merged sequence and such that t_i^x and t_j^y differ for no more than τ are paired, while events that have no neighbour within a time distance of τ are skipped; note that each event x can be paired with just one event y in the other sequence, and vice-versa.

For each pair of coincident events x and y with time occurrences t_i^x and t_j^y we set a synchronisation rate in the interval $[0, 1]$:

$$c_{i,j}(x, y) = \psi_\tau(d(x, y))$$

where d measures the temporal distance between events:

$$d(x, y) = |t_i^x - t_j^y|$$

and ψ_τ is a kernel depending on parameter τ, i.e., a decreasing function with finite support in interval $[0, \tau]$. Several kernels can be used, e.g., with exponential or sigmoid decay; we found a linear ramp to give the best results in our case:

$$\psi_\tau(t) = \begin{cases} 1 - \frac{t}{\tau} & \text{if} \quad 0 \leq t \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

In our experiments, we analyzed the impact of varying τ on the results obtained, and we show the findings on section 4.5.1 Similarly to [QKG02], the *average degree of synchronisation* Q_τ for pair of time-series ts_i and ts_j is finally given by:

$$Q_\tau = \frac{\sum c_{i,j}(x, y)}{(m_{ts_i} + m_{ts_j})/2} \tag{4.1}$$

Note that Q takes into account not only the number of synchronised events with respect to the number of occurring events, as in [QKG02], but it also provides a measure of the synchronisation strength in time, due to our weighted version of the synchronisation index $c_{i,j}$. The average degree of synchronisation associates a unique number in the interval $[0, 1]$ to each time series, which will be used in our experiments to rank the overall quality of performance.

4.5 Results

For all recordings described in Section 4.1 we computed the four PI series of clusters corresponding to arms and legs (see Section 2.3). Pairs of PI series from each trial were used next as input to the MECS algorithm. The result for each pair of series is a value Q_τ, in the range $[0, 1]$, where 0 means a total lack of synchronisation, while 1 means that all the detected events of the two limbs are perfectly synchronous (see Section 4.4).

Among all six possible pairs of PI series, we concentrate our analysis on two pairs: left arm vs right arm; and left leg vs right leg. Mixed combinations of an arm vs a leg were also tried, but provided less stable results; this is not surprising because many movements in karate involve just arms while legs remain static: in mixed combinations, a large number of events in one sequence find no relative event in the other, hence giving small values of Q_τ in all cases.

At the end, for each trial we obtain two values Q_τ^{arms} and Q_τ^{legs} to measure the level of synchronisation between arms and between legs, respectively. Since such indexes are on a congruent scale, we define the *Overall synchronisation index*

$$Q_\tau^{tot} = Q_\tau^{arms} + Q_\tau^{legs},$$

which will be used to run our statistical tests.

The purpose of our analysis is to show that the index Q_τ^{tot} provides enough information to discriminate the level of performance among to the three levels of athletes, as classified by experts. In the following, we first show that our index indeed characterises three different groups, and next we test the discriminative power of a simple classifier based on it.

4.5.1 Parameters setting

As described in Section 4.4, our method uses two parameters ρ and τ.

We have thresholded all our PI series with a value $\rho = 0.15$, which was found empirically to preserve the relevant peaks while discarding the residual noise.

Parameter τ determines the size of the kernel used to weight synchronisation. The value of Q_τ, hence Q_τ^{tot}, for the same data is monotonically increasing with τ. However, if τ is too small, too many pairs of potentially synchronous events are missed; while if τ is too large, the value of Q_τ^{tot} becomes less discriminative. For all trials, we carried out our analysis with multiple values of τ, to estimate the impact of this parameter on results. As summarised in the chart of Figure 4.4, the choice of τ is not critical: performances from athletes of higher level of skill return consistently higher values of Q_τ^{tot}; while such values remain discriminative on a reasonably large range of values of τ. From the chart, it appears reasonable to place τ in the range from 10 to 50 frames, i.e., from 1/25 to 1/5 of a second. In what follows, we set $\tau = 30$, i.e., about 1/8 of a second.

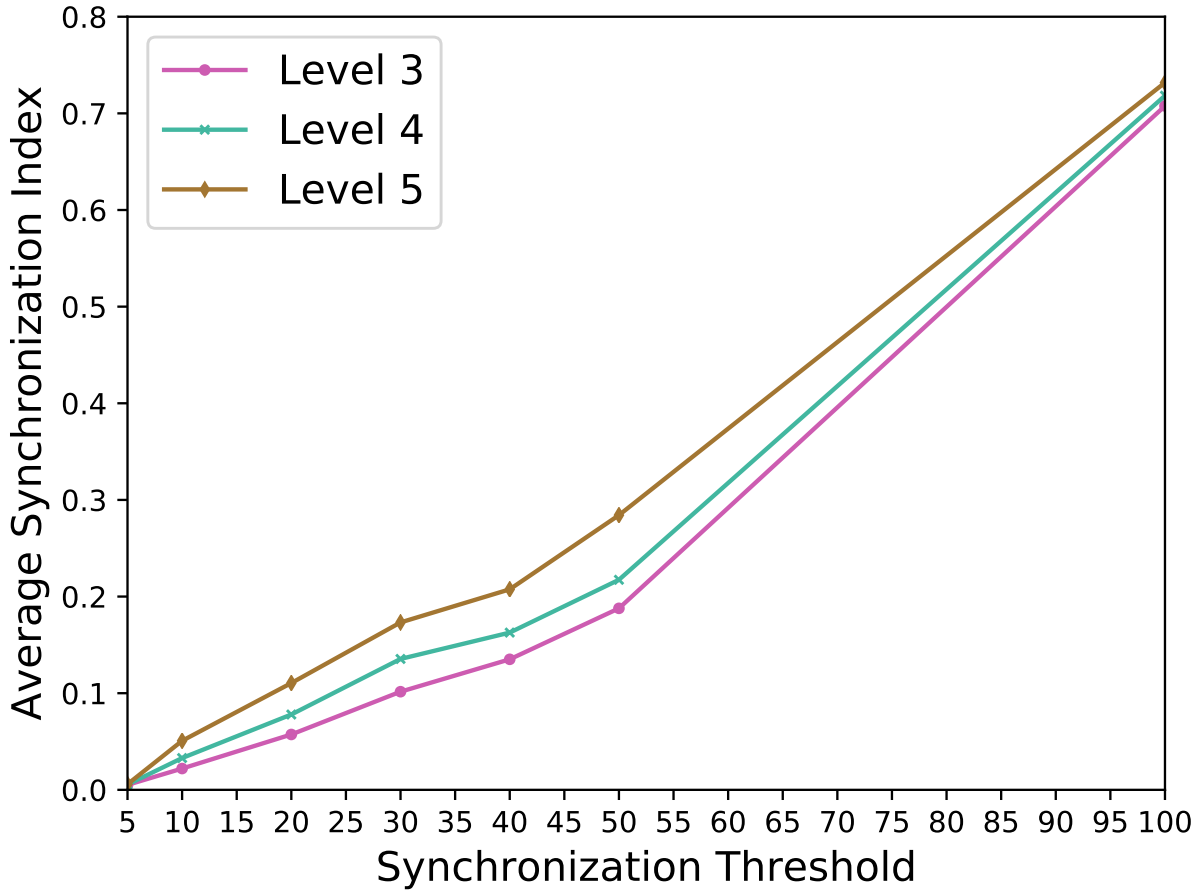


Figure 4.4: Average of the overall synchronisation index Q_{τ}^{tot} computed on the trials of each level at different values of the synchronisation threshold τ.

4.5.2 Statistical analysis

The barchart in Figure 4.5 shows the overall synchronisation indexes for all the trials, arranged in the three groups corresponding to the different levels of skill. Note that all scores for the group at Level 3 (magenta) are smaller than the scores of group at Level 4 (cyan), which in turn are smaller that the scores at Level 5 (brown). The visual analysis of this chart suggests that the overall synchronisation index succeeds in discriminating between different levels of skill. In order to provide a more rigorous evaluation of the discriminative power of our index, we have employed standard tools from statistical analysis.

We run tests aimed at discarding the null hypothesis, i.e., the assumption that our index characterises all data as coming from the same popualation. Since we are dealing with three different groups, we run a one-way ANOVA test followed by a Tukey HSD test [Nav06]. Both tests can be seen as extensions of the standard t-test to deal with more than two groups of samples, and they are often used together.

We first verify that it is reasonable to assume samples from each single group to come from a normal distribution (null hypothesis for each group), which is a necessary requirement to run the subsequent tests. Since our samples are generated by taking several takes of two different katas from several different athletes, the null hypothesis is all but a foregone conclusion. We have ran a Shapiro-Wilk test [SW65], which returns p-values of 0.635, 0.504, and 0.943 for Levels 3, 4 and 5, respectively. Such large values allow us to say that there is no evidence that the null hypothesis can be discarded, and suggest that each group may reasonably come from a normal distribution, but data are too few to support a stronger claim. However, also the Q-Q Plots depicted in Figure 4.6 show evidence that data are well fit by straight lines in all three cases, thus supporting the assumption that each group comes from a normal distribution. Therefore, we proceed under this hypothesis.

We first compute the mean and variance of each group to compare their distributions visually and numerically. Figure 4.7 shows the three Gaussians corresponding to such means and variances, which are clearly well distinct. We have computed the area of overlap of such Gaussians by using [ea14], in order to estimate the probability that an element is misclassified:

- The Gaussian of Level 3 overlaps the other two for just 0.0032, hence an element of Level 3 is classified correctly with probability 0.9968
- The Gaussian of Level 4 overlaps the other two for just 0.0032, hence an element of Level 3 is classified correctly with probability 0.9968
- The overlap of the Gaussian of Level 5 with the other two is below the precision of the method of computation that we have used; hence an element of Level 5 is classified correctly with probability $1 - \varepsilon$ where ε represents the numerical accuracy of the method.

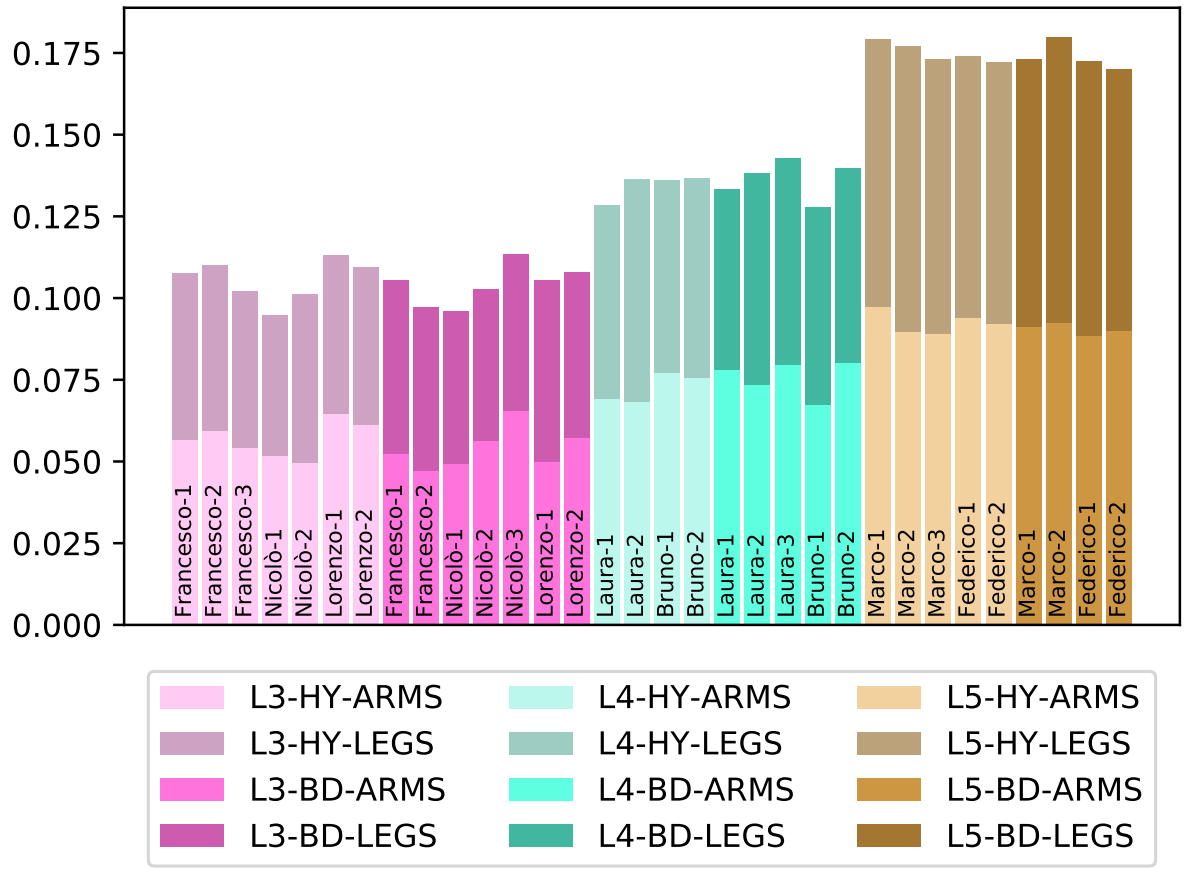
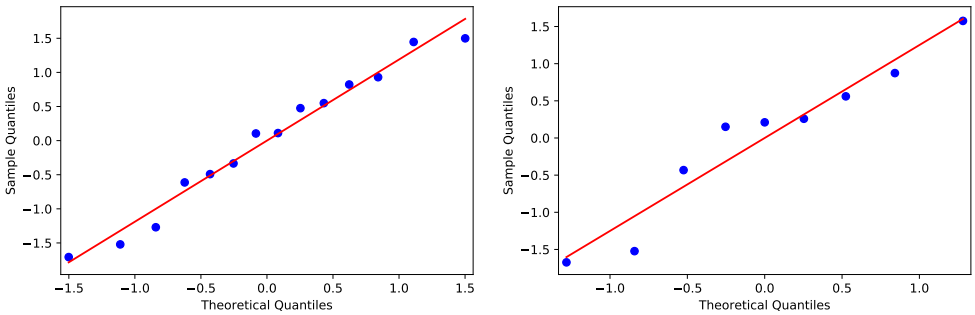
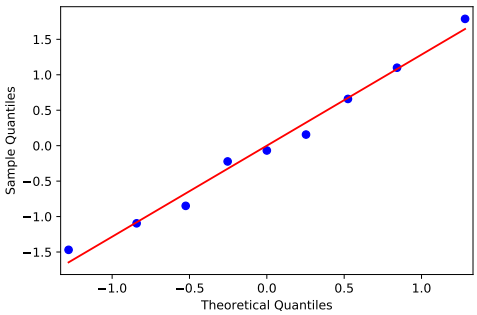


Figure 4.5: The overall synchronisation index for all trials in the dataset. Each bar represents the value of Q_τ^{tot} and is divided into two segments, representing Q_τ^{arms} (lower, lighter) and Q_τ^{legs} (upper, darker), respecitvely. Different hues correspond to the three levels (L3 magenta, L4 cyan, L5 brown). Less and more saturated colours correspond to the two different katas (Heian Yondan lighter; Bassai Dai bolder). Labels inside bars permit to identify different takes by the same subject.



(a) Level 3

(b) Level 4



(c) Level 5

Figure 4.6: Q-Q plots for the three groups at Levels 3 (a), 4 (b), and 5 (c) support the null hypothesis for each group.

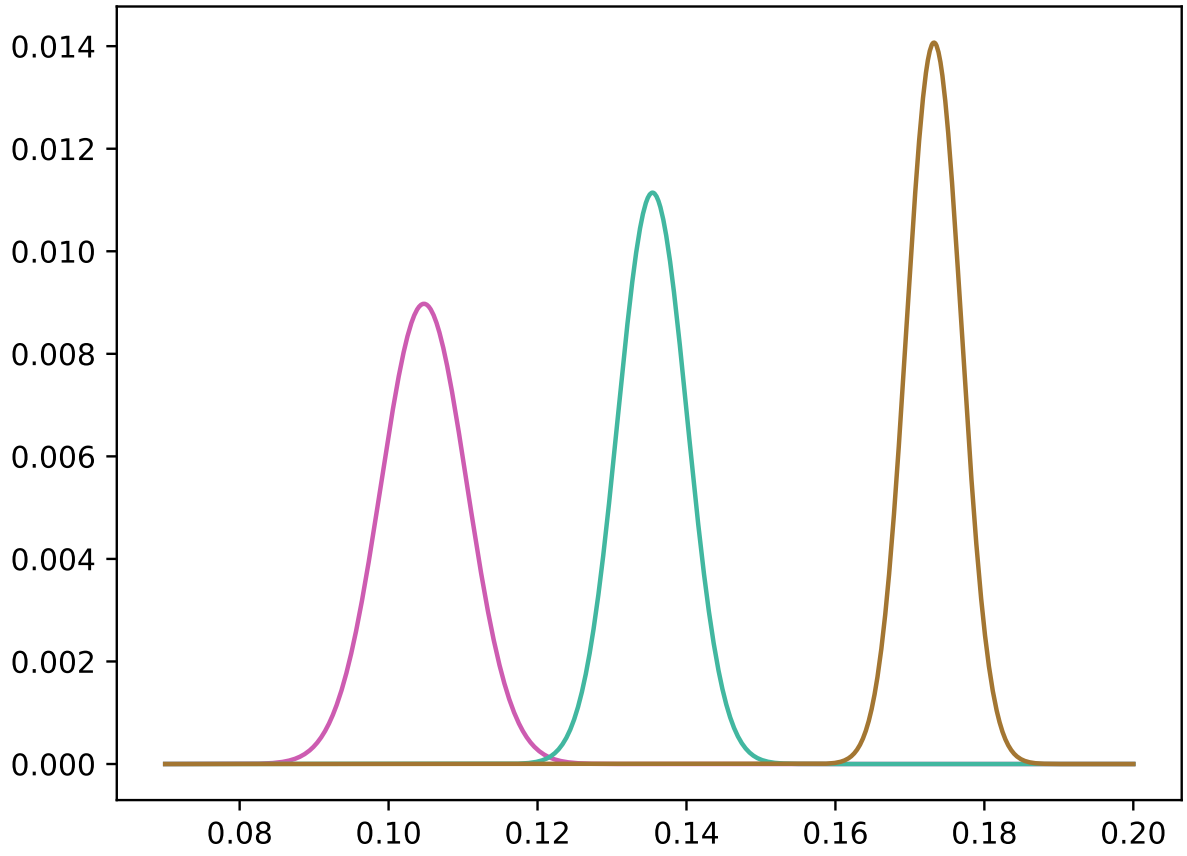


Figure 4.7: The Gaussians representing the ideal normal distributions of the three groups. The area of overlap below different curves gives the probability that an element is misclassified. Overlap between Level 3 and Level 4 is tiny; while the overlap between Lavel 5 and the other two groups is nearly null.

This result is further corroborated by the following tests.

The 1-way ANOVA test is aimed at assessing the statistical significance of means from different groups. Roughly speaking, it compares the inter-group variance with respect to the intra-group variance (a.k.a. F-test) and it computes the probability (p-value) that a value is obtained, which is greater or equal than the computed value under the null hypothesis for the whole set of samples (i.e., in case *all* the measured samples were from *the same* normal distribution). The p-value is compared with an alpha threshold that states the probability to reject the null hypothesis while it is true: any p-value smaller than the alpha threshold states the statistical significance of the samples, i.e., their means are statistically different and can be used to discriminate between different groups.

We set the usual significance threshold $\alpha = 0.05$, and we obtain a p-value $p < 10^{-12}$; since p is much lower than α, we can reject the null hypothesis and conclude that the means of the overall synchronisation indexes for the three groups are statistically different.

The Tukey HSD Test is another standard way to determine whether or not the different groups exhibit a significant difference. It compares the mean of every group to the means of every other group, and identifies any difference between two means that is greater than an expected standard error. Also in this case, a p-value is computed, which must be compared with the alpha threshold, with the same interpretation explained above. For all three possible pairs between Levels 3, 4 and 5, we obtain a p-value $p < 0.01$, meaning that the there is a significant difference between the means of the three groups.

In conclusion, both the ANOVA and the Tukey HSD tests confirm that our synchronisation analysis can be used to discriminate the three levels of skills, as classified by experts.

4.5.3 A basic classifier

On the basis of previous analysis, we define a basic classifiier as follows: let μ_3, μ_4 and μ_5 be the means of the Q_τ^{tot} indexes for the three groups of observed trials, respectively. Given a new trial, let q be its Q_τ^{tot} index. We classify the new trial to belong to:

- Level 3 if $q \leq (\mu_3 + \mu_4)/2$;
- Level 4 if $q > (\mu_3 + \mu_4)/2$ and $q \leq (\mu_4 + \mu_5)/2$;
- Level 5 if $q > (\mu_4 + \mu_5)/2$.

Better thresholds could be computed by taking into account also the variances of the three groups; however, our data are so well separated that we did not try a setting finer than the midpoint between consecutive means.

We test the performance of our classifier by cross validation through leave-one-out and bootstrap resampling. The leave-one-out technique culls one of the samples, say s, builds the classifier on the remaining samples and tries to classify s through it. We repeated the test by culling in turn all data and we obtained 100% of correct classifications.

The bootstrap resampling builds three new groups by random sampling from each group with replacement the same number of elements of the original group. Since resampling is made with replacement, some data are sampled multiple times, while some data remain out-of-bag. The classifier is built on the resampled groups, and it is tested on the out-of-bag data. We repeated the bootstrap resampling 100.000 times and we tested the resulting classifiers on a total of over 1 million data (out-of-bag). Also in this case, we obtained 100% of correct classifications.

4.6 Conclusion and Future work

We have presented a method to estimate movement quality in karate by studying how much the limbs are synchronised during relevant motion phases. Our approach demonstrates to be extremely robust on real examples consisting of MoCap data from 32 performances by athletes from three different levels of skill, as classified by experts in this martial art. A basic classifier built on our analysis succeeds in 100% of the subjects according to standard tests of cross validation. This suggests a strong correlation between the level of skill of an athlete and her ability to maintain a high intra-personal synchronisation in the movement of limbs.

Because of the lack of data, we could not test our approach on more than three classes. Given the results we obtained, we believe that there is good potential to apply the same approach also to several classes and to datasets in which the separation between consecutive classes is less sharp.

It should be noted that our analysis is completely independent from any assumptions on the input data: all the recordings are treated in the same way, independently of length, speed and specific performance. This makes our technique easily applicable with little effort to other scenarios in which we might want to measure the quality of a movement performed by different persons. One possible application is to measure the level of synchronisation of movements of dancers.

Events detected with our multi-scale approach could be used to derive also other measures of quality. For instance, kata usually contain many sudden and fast movements, and intuition suggests that experienced athletes are better at having a cleaner transition when starting the movement and, most important, when ending it. We plan to extend our analysis to measure the level of ”cleanness” of motion and use it as another way to discriminate between different levels of skill. We are also working on the application of the same analysis to the automated segmentation of different movements (e.g dance movements): relevant peaks of acceleration, as detected and ranked by the multi-scale analysis, give us a robust way to find the beginning and end of each relevant movements; also, the ranking among extracted peaks allows us to tune the granularity

of such segmentation.

Chapter 5

Segmentation Of Human Motion

Motion segmentation is the task of dividing a motion sequence into coherent elementary units. It is a common pre-processing step in feature extraction and motion analysis in various fields of applications including: animation, movement analysis for rehabilitation, choreography and dance, and robotics. Segmentation is performed either manually by users or automatically. In the latter case, manual segmentation is nevertheless needed to provide a ground truth, in order to evaluate the performance of automatic systems, hence the reliability of their generated segmentations.

5.1 Introduction

In our context, segmenting movement is the process of starting from a whole recording of a motion capture system, which might contain movements of arbitrary complexity/length/nature, and obtaining a set of basic movements which, combined, span the whole original recording (i.e. every instant in the original recording belongs to one and only one segment in the output). As it has been already discussed in chapter 3, there is a lot of variability inside mocap recordings: in an unconstrained setting where the performer is let free to execute whatever movement he/she likes at the desired speed there is virtually no constraint to how the final recordings will turn out: length, complexity of the movements, relevant features and size are all unknowns that makes analysis more complex.

Furthermore, the very definition of what constitutes a segment depends on the semantic problem addressed and on the particular features used; in the literature this problem has been widely studied for various kind of signals, such as audio and video, and it recently (approximately since the 2000s) some interest has been put into segmentation of motion capture data.

Given the complexity and variety of the problem, we try to develop a method that is as generic as possible, which makes no assumptions on the particular movements in the recordings: karate

movements as the ones described in 4 will have very specific basic movements making up the entire sequence (i.e. punches, kicks, and movements from switching from the end of one to the beginning of the other), while, for example, a flamenco dancer will have completely different kinds of movements, which will in turn be very different from a dancer dancing to a tradition Greek music.

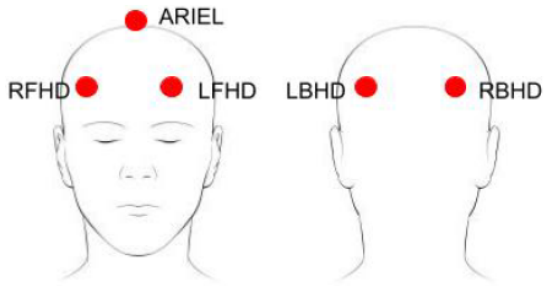
This generality induces also the need for the ability to extract different segmentations from the same recording: as stated before, the specific segmentation is dependent on whatever definition of segments is given. For example, we can see different granularity of segmentation in at least two different domains, the spatial one (i.e. are we considering each joint as relevant for the segmentation or just the whole body considered as a unique object?) and the temporal one (i.e. when considering the legs, is a step a segment? Or should we consider a segment just a series of steps starting and ending in a still position?). The goal is to have a segmentation that is able, in a single analysis, to extract and provide enough information to obtaining a parametric segmentation from which different subdivisions of the original recording can be obtained just by selecting a particular configuration of the parameters.

5.2 Datasets used

The first idea of developing a segmentation algorithm came while analyzing the dataset described in chapter 4, as karate movements can be a very simple testbed: they are clear, sharp movements, which follow precise rules and segments are well defined, as well as the points of transition between them. However, the dataset itself is quite small and noisy, and the availability of another dataset, more complex and of better quality, caused our research to shift on it.

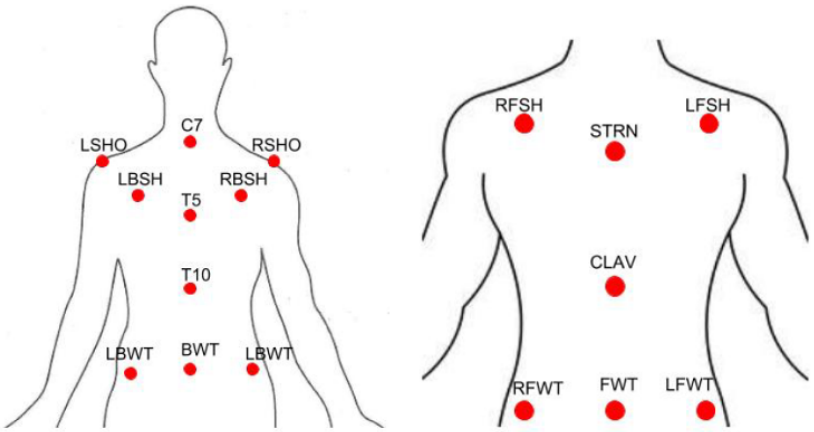
The dataset has been recorded under the European Union Horizon 2020 research and innovation programme under grant agreement No 688865 (WholoDance). It consists of a series of hundreds of recording (786 at the moment in which this is being written) of dancers tracked while performing different movements; the data contained ranges from very basic actions such as moving through the space recorded by the cameras tracing diagonals of an imaginary cube contained in it, to very complex sequences of dance steps, in some cases performed by two dancers tracked at the same time. Since this dataset contains much more variety to test the generality of our algorithm, we decided to focus on it; for reasons that will be specified in section 5.6, we selected a small subset of these recordings, focusing on the simpler ones.

The recorded data is similar to the one described in chapter 4, with passive markers being used to track one or more person performing; however, recordings are realized with a greater detail, because of improved hardware (more cameras are used with respect to the karate dataset) and an increased number of markers used. Indeed, in order to capture more subtle nuances of the movement, the number of markers has been increased from 25 to 64, as show in figures 5.1, 5.2, 5.3,5.4.



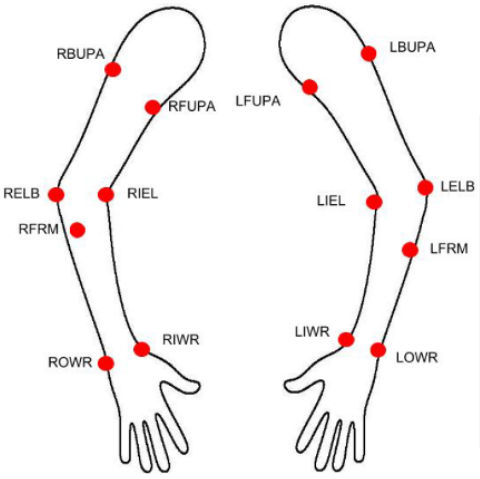
RFHD	Right Front Head
LFHD	Left Front Head
RBHD	Right Back Head
LBHD	Left Back Head
ARIEL	Top Head

Figure 5.1: Markers placed on the head

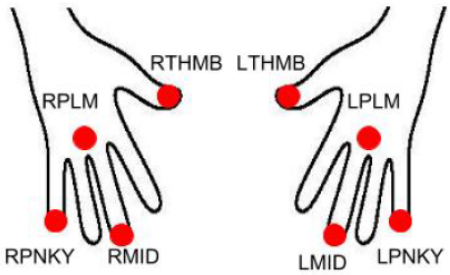


C7	7th Cervical Vertebra
LSHO	Left Shoulder
RSHO	Right Shoulder
LBSH	Left Scapula
RBSH	Right Scapula
T5	5th Thoracic Vertebra
T10	10 th Thoracic Vertebra
BWT	Sacrum Bone
LBWT	Pelvic bone left back
RBWT	Pelvic bone Right back
STRN	Sternum
CLAV	Xiphoid process
LFSH	Left Deltoid Muscles
RFSH	Right Deltoid Muscles
FWT	Below Navel
RFWT	Pelvic Bone Right Front
LFWT	Pelvic Bone Left Front

Figure 5.2: Markers placed on the back of the torso (on the left) and on the front (on the right)

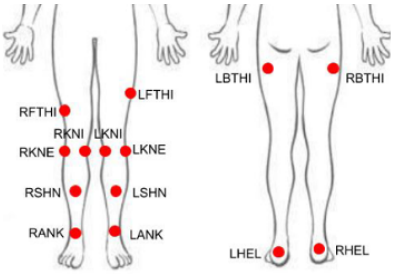


LFUPA	Left Front Upper Arm
LBUPA	Left Back Upper Arm
LIEL	Left Lateral Elbow
LELB	Left Medial Elbow
LFRM	Left Forearm
LOWR	Left Outside Wrist
LIWR	Left Inside Wrist
RFUPA	Right Front Upper Arm
RBUPA	Right Back Upper Arm
RIEL	Right Lateral Elbow
RELB	Right Medial Elbow
RFRM	Right Forearm
ROWR	Right Outside Wrist
RIWR	Right Inside Wrist

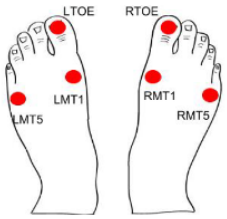


LPLM	Left Palm
LTHMB	Left Thumb Finger
LMID	Left Middle Finger
LPINKY	Left Pinky Finger
RPLM	Right Palm
RTHMB	Right Thumb Finger
RMID	Right Middle Finger
RPINKY	Right Pinky Finger

Figure 5.3: Markers placed on the back on the arms (on the left), and specifically on the hands (on the right)



LFTHI	Left Thigh Front
LBTHI	Left Thigh Back
LKNE	Left Knee (outer)
LKNI	Left Knee (inner)
LSHN	Left Tibia
LANK	Left Ankle
LHEL	Left Heel
RFTHI	Right Thigh Front
RBTHI	Right Thigh Back
RKNE	Right Knee (outer)
RKNI	Right Knee (inner)
RSHN	Right Tibia
RANK	Right Ankle
RHEL	Right Heel



LTOE	Left Toe
LMT1	Left 1st Meta Tarsal
LMT5	Left 5th Meta Tarsal
RTOE	Right Toe
RMT1	Right 1st Meta Tarsal
RMT5	Right 5th Meta Tarsal

Figure 5.4: Markers placed on the back on the legs (on the left), and specifically on the feet (on the right)

5.3 Feature selection

Feature selection is the first, crucial step to segmentation; the set of features that can be extracted from motion capture data is wide: from low level features (e.g. speed, acceleration, kinetic energy) to mid-level ones such as balance, equilibrium, fluidity and rigidity of the movement, up to most complex features like synchronization (see 4.4). However, given the generality of the problem we want to solve, we decided to remain at a low level, and to use just kinetic features. There is a number of reasons for using a low level feature: they are generic enough to make the system not dependent on any particular assumption, they are present and can be extracted in any kind of recording (while more complex ones might not make sense in some contexts) and they are good in discriminating relevant moments in the movement. In particular we will focus on the *acceleration*.

Given the high number of markers presented, a clusterization scheme has been devised; this time a hierarchical model has been selected, since the high granularity of the tracking make it possible to have different representation of the body skeleton at different levels of detail. From the cluster forming the whole body, three main clusters are defined, containing respectively markers of the Head, the upper and lower parts of the body; the cluster of the upper part is split into torso, left and right, while the lower part is just split into left and right; finally, from each arm a cluster containing just markers of the hand is defined, and analogously for the legs and the feet. The formal definition of the clusters is given in tables 5.1,5.2 and 5.3.

The process to identify relevant instants follows the same steps of the one described in chapter 4, with a special emphasis on the possibility of having multiple clusters hierarchically spanning the whole set of markers; once the subdivisions is decided, barycenters are computed and then their acceleration is extracted.

We employ a filtering technique analogue to the one shown in chapter 4 to obtain a smoothed computation of the velocity of the barycenters; again in this case we focus on critical points of the acceleration, which have the semantic meaning of being points where movements start or

UPPER BODY	C7	LSHO	RSHO	LBSH	T5	RBSH	**TORSO**
	LBWT	BWT	RBWT	RFSH	STRN	LFSH	
	RFWT	FWT	FWT	T10	CLAV		
	RBUPA	RFUPA	RELB	RIEL			**UPPER RIGHT**
	RFRM	ROWR	RIWR				
	RTHMB	RPLM	RPNKY	RMID			
	LBUPA	LFUPA	LELB	LIEL			**UPPER LEFT**
	LFRM	LOWR	LIWR				
	LTHMB	LPLM	LPNKY	LMID			

Table 5.1: Clusters for the upper part of the body. Markers with a colored backgrounds constitutes clusters of the hands (green for the right hand, orange for the left one)

LOWER BODY	RFTHI	RKNE	RKNI	RSHN	RANK	**RIGHT LEG**
	RBTHI	RHEL	RTOE	RMT1	RMT5	
	LFTHI	LKNE	LKNI	LSHN	LANK	**LEFT LEG**
	LBTHI	LHEL	LTOE	LMT1	LMT5	

Table 5.2: Clusters for the lower part of the body. Markers with a colored backgrounds constitutes clusters of the feet (green for the right feet, orange for the left one)

HEAD	ARIEL	RFHD	LFHD	RBHD	LBHD

Table 5.3: The composition of the head cluster

end (in the first case we have a maximum of acceleration, in the other one a peak of deceleration, which is a minimum of the acceleration signal).

5.4 The concept of scale in motion segmentation

The usefulness of multi-scale approach has already been discussed in chapter 2 and shown applied to a real problem in chapter 4. Everything that has been said before still holds in this case, but the concept of scale is even more relevant in the problem of motion segmentation, as it can be considered in two different domains:

5.4.1 Space

As discussed before, the recordings are realized by tracking a large number of markers (64), spanning the whole body. The hierarchical clustering gives us not only a way to employ redundant information and smooth the effect of eventual noise present, but also a control over the granularity of the segmentation in the space domain. This gives us the power to answer the question *"What is a segment?"* in different ways according to the clusterization chosen. In particular, we can answer that a segment is a movement performed by the whole body, or just by one arm, down to the level of detail of saying that even if the whole body is still and just the right hand is moving, we consider it a segment.

5.4.2 Time

Perhaps the most intuitive relation between segmentation and the concept of scale is related to the temporal domain. Let us suppose we fixed the space granularity of our segmentation by analyzing, let us say, the right arm. The question *"What is a segment?"* still leaves us with many options, for example:

- Movements that start and end at a still position: this is the coarsest level of segmentation.
- *Coarse* segments can be further divided into movements that compose them; this is realized by looking inside the segment and extracting finer features.
- The finest level of segmentation is the one in which every peak of acceleration constitutes the end of a segment and the beginning of a new one.

It should be noted that, while the example gives a *discrete* choice on the temporal granularity (coarse, medium, fine), the choice is actually determined by a parameter that is continuous, giving even more control.

5.5 Extracting the segmentation

The multi-scale analysis is carried out analogously to how described in 4, combined with the classical scale-space analysis, to obtain two congruent signals, that have the same length of the input one and also have non-zero values only at the location of its critical point; as described in section 2.4

The segmentation is then extracted via a thresholding on the resulting signal; the threshold parameter gives us control over the granularity of the segmentation. Each instant that is over the selected threshold is considered a point that divides two segments; this gives us a full segmentation (i.e. every frame is part of a segment) with no overlap. Lowering the threshold creates more divisions, thus computing a finer segmentation. Note that we define segments as sequences of frames between two relevant instants, and not with a semantic definition on the segment itself; this would make the creation of a ground truth easier, as the process should relay only on the identification of single frames instead that on the analysis of whole periods of time.

5.6 Preliminary Results

To the best of our knowledge, there is a lack of mocap dataset adequate for testing our algorithms, so we ran an empirical validation done visually by non-expert in the dancing field; given the fact that preliminary results seems promising, we decided to proceed with the collection of ground-truth data.

We chose a subset of the recordings discussed in 5.2, consisting of basic movements that are not related to any dancing style, and then some basic steps of three types of dance: greek, flamenco, contemporary. The choice of relatively few recordings is due to the fact that to have a reliable ground truth we prefer having many annotations of just some recordings rather than single segmentations on many recordings. We will discuss the problem of having a reliable ground truth later in the section.

We set up a web based tool which allows users to annotate recordings and mark the instants in which they see a phase of *transition* between two segments. Users can go frame by frame in the video, thus having the most accurate control on their annotations, which they can also later rewatch and edit if needed.

Table 5.4: Average of synchronization of the manual segmentation with two different kernels

	0.2s	0.5s	0.8s	1.s	1.2s	1.5s	2s
Linear Kernel	0.1599	0.2161	0.2594	0.2799	0.2978	0.3224	0.3571
Uniform Kernel	0.2041	0.2973	0.3579	0.3824	0.4027	0.4361	0.4845

5.6.1 Analysis of ground truth data

With the aforementioned tool, we were able to collect ground truth on 26 recordings, each one segmented manually by two experts in the dance field. We ran the synchronization algorithm discussed in Chapter 4 with two different kernels (uniform and linear) to analyze how much the manual segmentation are in sync one with the other, with the key concept that a synchronization of 1 will mean that the two dance experts were completely in agreement on the segmentation. Table 5.4 shows the average value of the synchronization index with different values of τ (in seconds, since not all the recordings have the same framerate).

As we can see from the table, the results show that manual segmentation is still an issue, in the sense that, even with a 2 seconds window and an uniform kernel (which is the one that always gives higher synchronization), there is consensus, on average, only on half of the segmentations extracted. Since the number of samples is right now too limited to conclude anything at this point, the process of manual annotation is still ongoing, with the aim of constantly analyzing the results to, eventually, get to an higher agreement between experts.

Chapter 6

Scale-Space Techniques for Fiducial Points Extraction from 3D Faces

In this chapter we propose a method for extracting fiducial points from human faces that uses 3D information only and is based on two key steps: multi-scale curvature analysis, and the reliable tracking of features in a scale-space based on curvature. Our scale-space analysis, coupled to careful use of prior information based on variability boundaries of anthropometric facial proportions, does not require a training step, because it makes direct use of morphological characteristics of the analyzed surface. The proposed method precisely identifies important fiducial points and is able to extract new fiducial points that were previously unrecognized, thus paving the way to more effective recognition algorithms.

References:
Nikolas De Giorgis, Luigic Rocca and Enrico Puppo. 2015. Scale-Space Techniques for Fiducial Points Extraction from 3D Faces. In Proceedings of the 18th International Conference on Image Analysis And Processing (ICIAP 2015).

6.1 Introduction

Face recognition is a widely known problem, which has been addressed by many authors in literature, mainly in the image processing field for many years. Given a test image representing a face and a database, the goal is to obtain an algorithm that permits to identify in the database the face image that represents the same face as the test one. Most of the work in the past has been developed to work with color and gray-scale images (worth citing are the most famous: PCA [HSE03], LDA [Fri89], and EBGM [US05])). Recognition from 2D images, though, suffers from several known problems, such as strong dependences on illumination and pose; images also

lose most of the information about the structure of the face, which is inherently 3-dimensional, by projecting everything into one plane. Despite these shortcomings, works on 3D face recognition are way less present in the literature, since acquisition hardware for 3D data were extremely rare and expensive until a few years ago, and as consequence 3D face datasets were almost non existent. In the last few years, though, manufacturer worked towards way cheaper and easier to use acquisition hardware, and the ever growing increase of computation power available even with small budgets unlocked the demand and, consequently, the production of high level software to deal with every phase of the acquisition workflow, such as surface reconstruction, meshing, cleaning and smoothing. It also opened the door to completely new techniques for the extraction of 3D data, such as the employ of photogrammetry to extract precise 3D meshes of faces from a set of photos.

Existent methods working with 3D data can be roughly subdivided into two categories:

- *appearance based* algorithms are usually modified versions of 2D methods that are extended to work with range images.
- *feature based* algorithms define some sort of measure defined on the input face and then extract landmarks that are locally relevant with respect to this measure.

The work we developed falls into this latter category, and focuses on the extraction of a subset of points defined in the medical literature, called *fiducial points*

6.2 Fiducial Points

There have been numerous works in the medical literature about identifying and describing facial features that are able to describe the whole set of possible variations of facial characteristics; this whole research field falls onto the definition on cranio-facial anthropometry and dates back to at least three centuries ago. Cranio-facial proportions between set of points on the human face have been used widely in a number of areas: in anthropology to analyze prehistoric human remains ([Com]), for quantifying facial attractiveness ([Far58]), as an aid to cosmetic and reconstructive surgery ([FM87]). The seminal work in the field by Farkas and Munro [FM87] introduces a list of 155 cranio-facial proportions physical measured on 2564 persons belonging to various ethnic, gender and age groups ([FM87]). Among these 155 proportions, 70 of them are completely related to the facial area, and Gupta et al. ([GMB10]) extracted the 23 associated with the highest standard deviation values as the representative of the most discriminatory facial structure characteristics.

The set of facial landmarks which spans the 23 proportions is made of 25 points that are shown in figure 6.1.

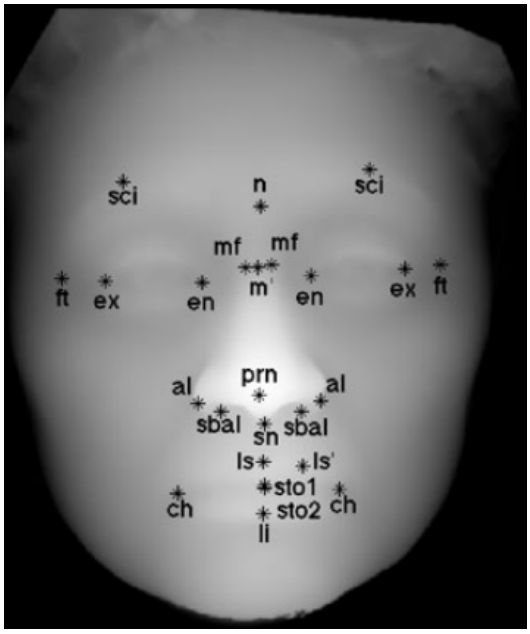


Figure 6.1: The set of 25 facial landmarks employed by Gupta et al to compute the facial proportions

6.3 State Of The Art for Feature Based Techniques

Now, wer are going to provide an overview of the feature based techniques have been developed in the past year. Lu And Jain [LWJ03] proposed a methods that combines 2D and 3D techniques to extract a small set of facial features: they use *a priori* knowledge and pose estimation to extract the tip of the nose, then they detect the mouth and eye corners using the shape index from the range image and corner extraction from the intensity image. Gupta et al. [GMB10] developed a method which detects a set of 10 points (subset of the *fiducial* points shown in figure 6.1), combining information about surface curvature and 2D techniques (edge detection) and they need to have both the intensity image and the range image (aligned). A recent work by Perakis et al [PPTK13] uses the shape index and spin images to extract a small set of 8 points under large yaw and expression variations. Spin images were used also by Conde et al. [CCRa$^+$] with high accuracy but restricted to just three points of the face (The tip of the nose and the inner corners of the eyes). Segundo et al [SSBQ10] used techniques similar to the ones we will show in the next sections (surface curvature and depth information from range images) but limited to a small set of points (nose tip, nose and eye inner corners); a paper by Shin and Sohn [SS06] utilizes ten facial landmarks in the recognition problem but give no information about how them are extracted. Sukno et al [SWW12] use the concept of spin images as tool for object recognition in 3D as shown in [JH99] to extract some facial landmarks but then make massive use of statistical models to filter out outliers and infer missing feature. Bockeler and Zhou [BZ13] extract ten

points mainly relying on 2D information and anthropomorphic constraints. A work by Berretti et al [BWdBP13] computes DoG of a mean curvature scalar field to extract a variable number of keypoints relevant for the field defined but not necessarily located in meaningful parts of the face. Finally, some works by Novatnack et al [NN07, NN08, NNS06] use mesh parametrization with a distortion-adapted Gaussian scale-space to extract features using techniques from image processing such as edge and corner detection.

Our contribute extends the family of techniques based of curvature and priori knowledge of antrophometric feature's location, but it relies only of information coming from 3D surface, freeing it for well-known problem about color or light intensity and direction; it also requires no learning or training.

6.4 Surface Curvature

6.4.1 Background

We will give here some basic notion of differential geometry, which will then be used throughout the chapter to characterize fiducial points. Let S be a smooth surface and let $N_S : S \mapsto \mathbb{R}^3$ be its normal field (also known as *Gauss Map*), i.e. a vector field associating to each point $P \in S$ its surface normal $N_S(P)$; the *shape operator* is then defined as the negative differential of the Gauss map:

$$S = -dN_S$$

The shape operator associates to each point $P \in S$ a linear operator which describes how the normal vector changes along any direction on the tangent plane of S at P; it can be described at each point P by a 2×2 matrix S_p relative to a local frame $(\mathbf{u}, \mathbf{v}, \mathbf{n})$, with origin in P and such that $\mathbf{n} = N_S(P)$

Eigenvalues k_1 and k_2, and eigenvectors $\mathbf{t_1}$ and $\mathbf{t_2}$ of S_p are the values and the directions, respectively, of the *principal curvatures* of S at point P. The directions $\mathbf{t_1}$ and $\mathbf{t_2}$ are mutually orthogonal and lie on the plane $T(P)$, which is the tangent plane of S at point P, while k_1 and k_2 are the maximum and minimum values of this curvature (k_1 is the curvature along direction t_1, and the same holds for k_2 and t_2). The product $k_1 k_2$ is called the *Gaussian curvature*, and it is denoted with K.

Values of the principal curvature are used to classify points on the surface, and specifically concavity and convexity of the surface, as shown in table 6.4.1

	$\mathbf{k_1 < 0}$	$\mathbf{k_1 = 0}$	$\mathbf{k_1 > 0}$
$\mathbf{k_2 < 0}$	Concave ellipsoid	Concave cylinder	Hyperboloid surface
$\mathbf{k_2 = 0}$	Concave cylinder	Plane	Convex cylinder
$\mathbf{k_2 > 0}$	Hyperboloid surface	Convex cylinder	Convex ellipsoid

Table 6.1: Classification of surface points based on their principal curvature values

In this work we will though mostly use the Gaussian curvature K to characterize surface points in the following way:

- If both the principal curvatures are of the same sign, then $K > 0$ and the surface is said to have an elliptic point. The surface will be dome-shaped (either concave or convex)
- If the principal curvatures have different signs, $K < 0$ and the surface is said to have an hyperbolic or saddle point (from the shape the surface will have, see fig 6.2)
- If one of the principal curvature is equal to zero, then also $K = 0$ and the surface is said to have a parabolic point

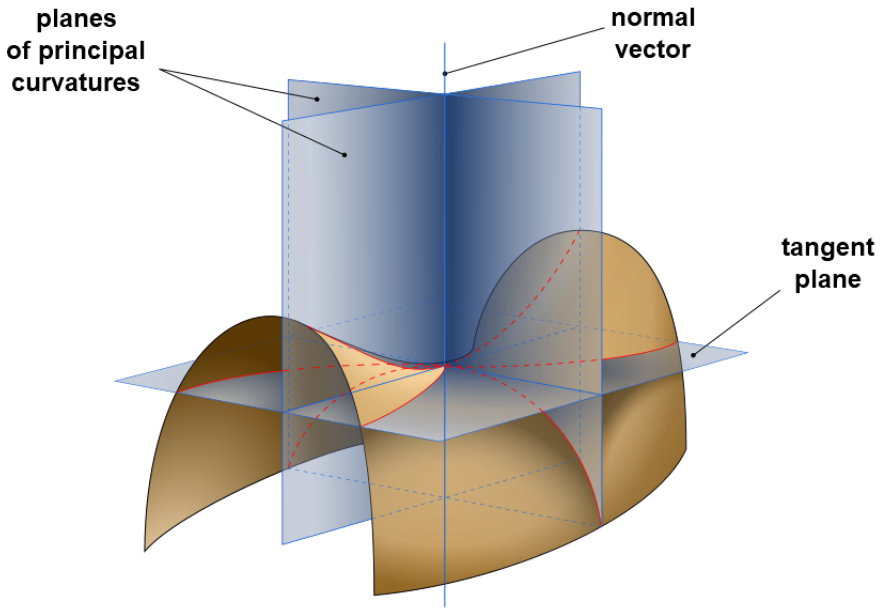


Figure 6.2: A saddle surface

6.5 Fiducial Points Characterization

Gaussian curvature, as described in the previous section, gives a strong characterization of surface point; in particular, among the 25 points identified in 6.1, the 13 points shown in 6.3 are the ones that are more precisely characterizable by their Gaussian curvature (along with other features that will be described later). In particular, we can see that most points are ellipsoids, either concave (ex_l,ex_r,el_l,el_r,ch_l,ch_r), or convex (prn,li,ls) while others are hyperbolic points (m, sn,al_l,al_r). For some points this might not seem obvious, as while some can be relevant in a local frame (e.g. corners of the eyes) others have distinctive curvatures only at a larger scale (i.e. m, the *nasion*). For this reason our works relies on a multi-scale representation and analysis of the curvature.

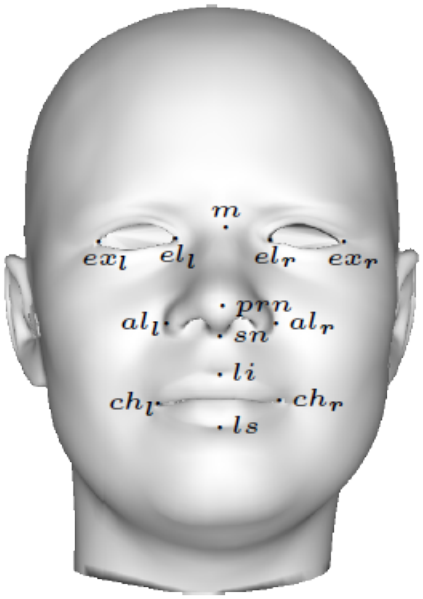


Figure 6.3: The set of 13 fiducial points extracted by our method shown on one of our input meshes

6.6 Method

6.6.1 Curvature Extraction

While most of the *feature based* algorithms using curvature compute it with a discrete method, that is known in the geometry processing field to be prone to noise, with an increased effect especially on datasets of bigger size and resolution. We rather employ a multi-scale curvature

analysis method based on surface fitting [PPR10]; we will give here just a small summary of how the method works emphasizing is relationship to the concept of *scale* which is the main ingredient of this work. The computation of the curvature at a vertex v of a mesh m is extracted by gathering a local neighborhood of v; then a local frame is set centered on v and with the average of the neighborhood's normals as one axis and the directions spanning the tangent plane of m at v as the other two. The neighborhood is then express in this local reference frame, and then a polynomial of degree two fitting this data is computed. Knowing its formula, the curvature is extracted by computing the shape operator at point $(0, 0, 0)$.

The size of the local neighborhood is an important parameter that can be chosen by the user, and it has a relevant consequence on the extraction of the curvature: a small size implies that each vertex contributes more both to the computation of the normal and to the definition of the fitting polynomial; thus a small neighborhood implies that each vertex is more relevant, and details are better preserved. Conversely, a large neighborhood makes each vertex less relevant, and the resulting fitted surface will be spread amongst more points so that the contribution of each one of them will be lower, thus making smaller details disappear.

This behavior is highly similar to what happens along scales of the scale-space approach (Section 2.2), and the combination of the two approaches will be described in the next section.

6.6.2 Diagonal Scale-Space

Furthermore, in order to get a ranking of critical points of Gaussian curvature, we employ scale-space techniques described in chapter 2; in our case, instead of having an input signal which gets repeatedly smoothed with a filter, we have a collection of samples of the scale space obtained by varying the scale parameter of the surface fitting algorithm described early. Computation of differential properties, though, it is severely affected by the presence of noise, since curvature is computed from second order derivatives. This makes the most straightforward implementation of the scale-space, the one just described,not adequate for the detection of relevant features, since the number of critical points does not decrease fast enough as the scale parameter increases for the tracking to provide meaningful information.

We thus propose a novel implementation of the scale-space which combines multi-scale curvature computation and Gaussian scale-space, called *diagonal scale-space*.

In our case the smoothing process on the original surface is carried out with a Gaussian smoothing of range images, but the general idea of two parameters controlling the smoothing on different domains (original signal and extracted feature) can be applied also in other contexts (e.g. triangular meshes smoothed with a Laplacian filter). The end result is that noise is discarded in a more effective way, critical points disappear faster through scales and we obtain a more meaningful tracking of feature across scales.

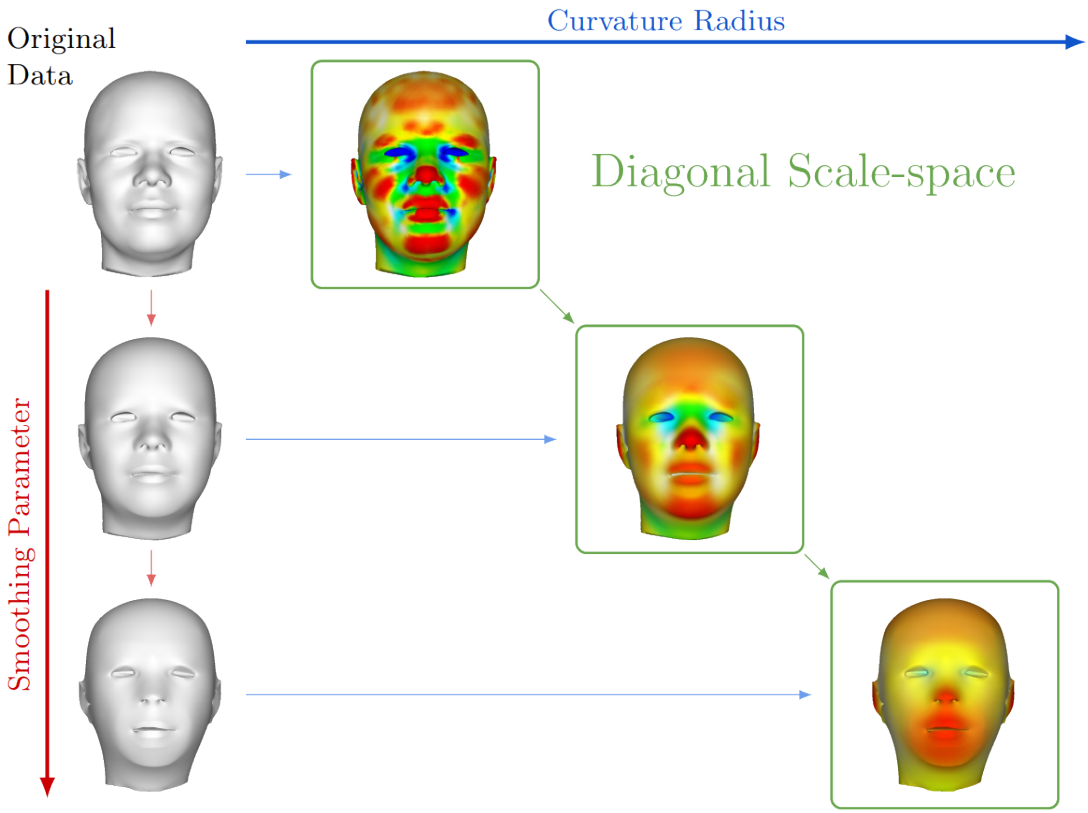


Figure 6.4: The diagonal scale-space is composed by a sequence of curvature fields, obtained by computing curvature at increasing scales on increasingly smoothed surfaces.

6.6.3 Critical Points' Importance Measure

After generating the diagonal scale-space described before, we extract all critical points in the original signal (in our case, this is the curvature computed on the original, non-smoothed surface, at the smallest radius) and we track them through scales using the virtually continuous scale-space method described in [RP13]. Its output is a data structure which encodes every critical point with detailed information about how it changes through scales. From this structure we can extract two measure of importance associated to critical points.

6.6.3.1 Life

Life is the most intuitive measure of importance that can be associated to critical points; since the data structure memorizes each event that occurs to each critical point, we exactly know the moment in which disappears after being smoothed out. This event marks the lifetime of the critical point, and we use this *life* value as our main measure of importance.

6.6.3.2 Strength

As a secondary measure of importance we compute the relative strength of the scalar field's maxima and minima compared to the local trend on the surrounding surface. For each maximum, , we compute the average of the curvature field at the pixels that are below its value in a growing area around it, and return the highest difference between its value and that average; the same algorithm is applied to minima by taking into account only the surface values above the minimum. The radius of the local area is capped at a value related to the scale of its life in the scale-space. The resulting value corresponds to a sort of variable-scale Laplacian of the surface at a given point.

A graphical depiction of critical points scaled by either life or strength is shown in figure 6.5

6.6.4 Extraction of fiducial points

As previously discussed, fiducial points are selected among maxima and minima of the Gaussian curvature scalar field. We employ a strategy that relies only on prior knowledge about the human face and on the *life strength* measure described in the previous section.

We follow a hierarchical search, starting from the most prominent features and then restricting our search to narrower areas based on the displacement relative to the previously found ones. In particular, we start from fiducial points that characterize the noise, then we compute the symmetry axis that separated the left and right part of the face; we finally locate the other points.

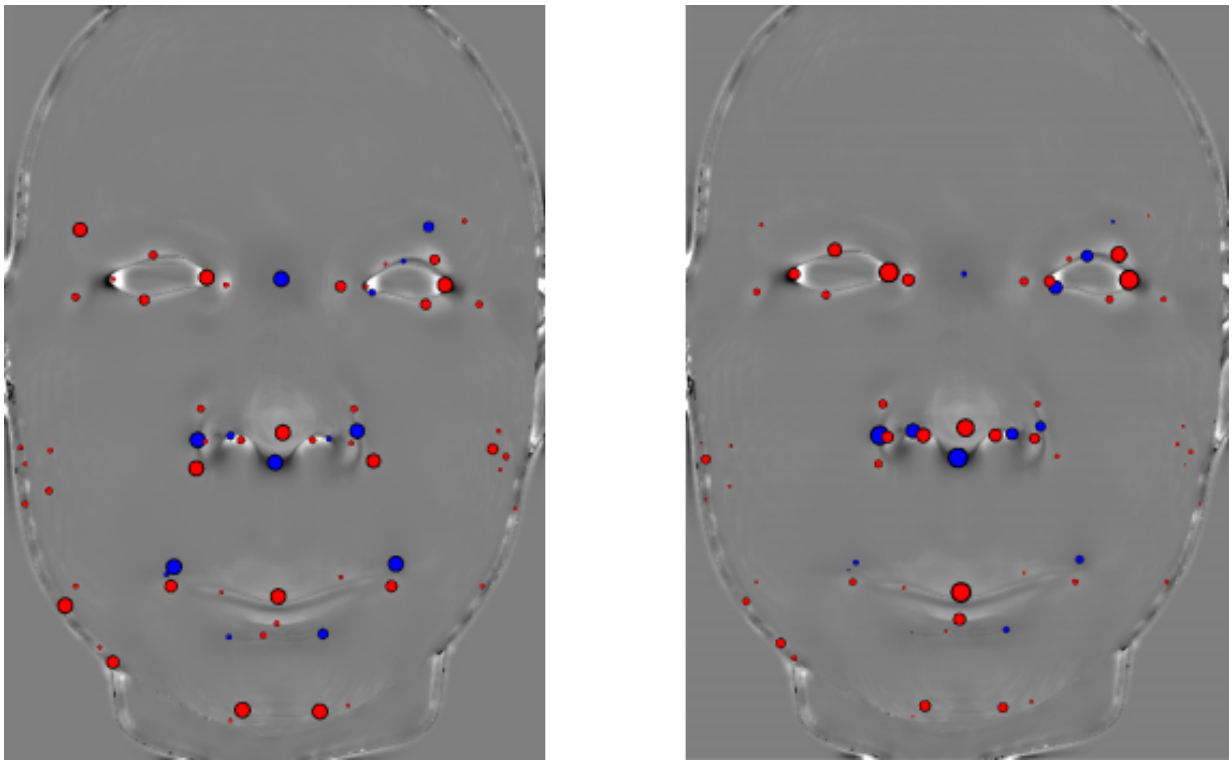


Figure 6.5: Maxima (red) and minima (blue) of Gaussian curvature scaled by life (on the left) and strength (on the right).

6.6.4.1 The nose

There are five fiducial points that characterize the nasal area (they are shown in figure 6.6a), namely prn (the nose tip), al_l and al_r (the sides of the nose), m (the upper saddle) and sn (the lower limit of the nose).

- prn: this fiducial point is characterized by a very high Gaussian curvature and a very long life in the scale-space process; we found out that the best strategy to locate it is to restrict the search to a rectangular area around the center of the image (shown in red in figure 6.6a) and to find the point with the highest Gaussian curvature at the last level in which its related critical point is still alive.
- al_l,al_r: they are the lateral sides of the noise; on the surface they appear on saddle points, meaning we expect them to be minima of Gaussian curvature; in fact, we locate them by scanning left and right to the nose tip selecting minima with the highest life in the scale-space.
- m: this point is a very prominent saddle on the surface, even at a coarse scale. In fact we can see that the minimum of Gaussian Curvature related survives very long the smoothing process, and we extract it by looking at a rectangular portion of the face above the nose tip and selecting the minimum with the longest life.
- sn: the lower limit of the nose is again a saddle; we have discovered that this point is best characterized by the *strength* value, so we look in the rectangular area below the nose tip and select the point with the highest strength.

It should be noted that bounded areas to restrict the search has been used to both improve accuracy (they reduce the chance to get false positive) and speed up computation; however, their size have been decided by following very well known facial proportions and have been deliberately kept a bit larger than needed to not too heavily rely on them.

6.6.4.2 Symmetry axis of the face

After extracting the points around the nose, we use them to estimate the vertical symmetry of the faces; the goal is to take advantage of the intrinsic symmetry of the human face and the fact that most of the remaining fiducial points are both on the left and right side. It is computed in the following way:

- Firstly, we compute the line l_1 that is the best linear fit of the points m,prn,sn.
- We also compute the line that connects points al_l and al_r, and the the line l_2 as its orthogonal line.

- We finally compute the symmetry line as the average between l_1 and l_2.

A graphical depiction of the symmetry axis and the line connecting al_r and al_r (which will be used to distinguish between higher and lower regions of the face) is shown in 6.6b.

As said before, most of the points that still need to be detected are symmetric pairs with respect to this axis, so when looking for them we will search for both points at once, introducing the requisite for them to be *almost* symmetrical (up to a certain tolerance) on top on any other criteria that might be necessary in order to identify them.

6.6.4.3 The Eyes

The identification of the corner of the eyes is shown in the upper half of Figure 6.6c: the two pairs of external (ex_l and ex_r) and internal (en_l and en_r) corners are located in pit regions with very high Gaussian curvature; we extract the two symmetric pairs in the upper half of the image with the highest strength value: we detect all possible symmetric pairs of (a, b), with strength values (s_a, s_b) and we select the two pairs that maximize the value $s_a \cdot s_b$

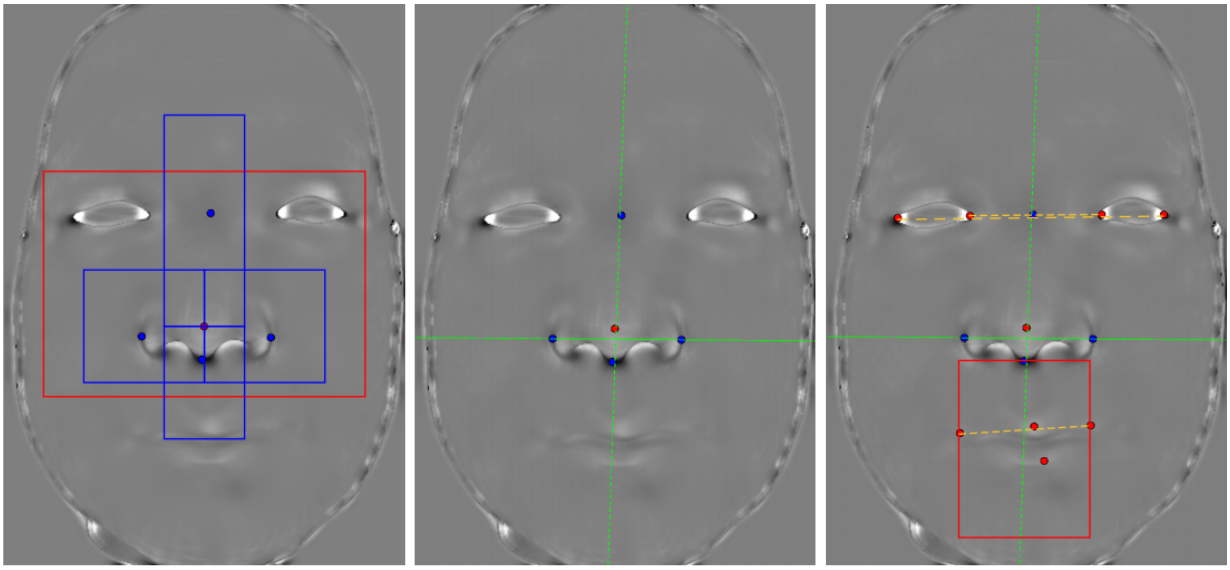


Figure 6.6: (a): The first five points and the bounding boxes used to find them. (b): The horizontal line across al_l and al_r divides the face in an upper half and a lower half; the vertical line represents the symmetry axis computed on the given face. (c): Remaining points located through symmetric search, connected by a yellow dashed line, plus bounding boxes for points ls and li.

6.6.4.4 The Mouth

This area contains for fiducial points (see the lower half of Figure 6.6c): two of them are part of a symmetric pair (the corners of the mouth) while the other two represent the tip of the higher and of the lower lip. The corners of the mouth ch_l and ch_r are identified with the same strategy employed for the eyes' corners, applied this time to the lower half of the face. The upper lip ls and the lower lip li are identified as the two maxima of Gaussian curvature with the highest life in an area located below sn and delimited by ch_l and ch_r (shown in Figure 6.6c).

6.6.5 Results

We tested our method on the Face Warehouse Dataset [CWZ+14], using meshes representing faces with neutral expressions, frontally projected in order to extract range images, for a total of 111 different faces. Since the dataset does not provide a ground truth for fiducial points, we manually created it by selecting the 13 points shown in figure 6.3 on each image.

We evaluated our method by measuring the distance in millimeters between each fiducial point extracted by our method and the corresponding point in the ground truth dataset, for each mesh. Plots in Figure 6.7 show the percentage of meshes (on the Y axis) on which the distance is less than the given one (on the X axis), for each fiducial point.

- Figure 6.7a shows the results for the detection of points characterizing the nasal area; the accuracy for these points is high; at a distance from the ground truth of 4mm, the tip of the nose *prn* is localized on 99% of the dataset, while *sn* is localized on 94% of the dataset. To the best of our knowledge, this work is the first to use 3D methods to detect this point. At 7mm of tolerance, the sides of the nose al_l and al_r reach a detection rate of 90%. The worst performance in this area is achieved on detecting m, the nose saddle, which reaches 90% at 11mm. It should be considered that it might by partly due to the difficulty in manually placing this point, since the nose saddle is a wide area and it is not always clear where the exact location of m should be

- Figure 6.7b shows the results of the eyes' corners. Our methods performs with good accuracy for all these points: all point reach a detection rate above 90% within a 4mm distance, and the inner corners en_l, en_r reac 99% at 4mm.

- Figure 6.7c shows results for fiducial points located around the mouth; features related to these points are subtle, close the noise level, thus heavily affected by it. In fact, only a few works have tried to detect the mouth corners ch_l and ch_r [BA00, BZ13, GMB10, PPTK13, SWW12], and always by incorporating 2D information. In our case, the extraction suffers the fact that a lot of points along the mouth tend to have similar curvatures. The accuracy of 90% is reached at 14mm. To the best of our knowledge, this work is the first to perform

3D detection of the fiducial points on the upper and lower lip, namely ls and li. Detection for these points achieve a 90% rate at 8mm.

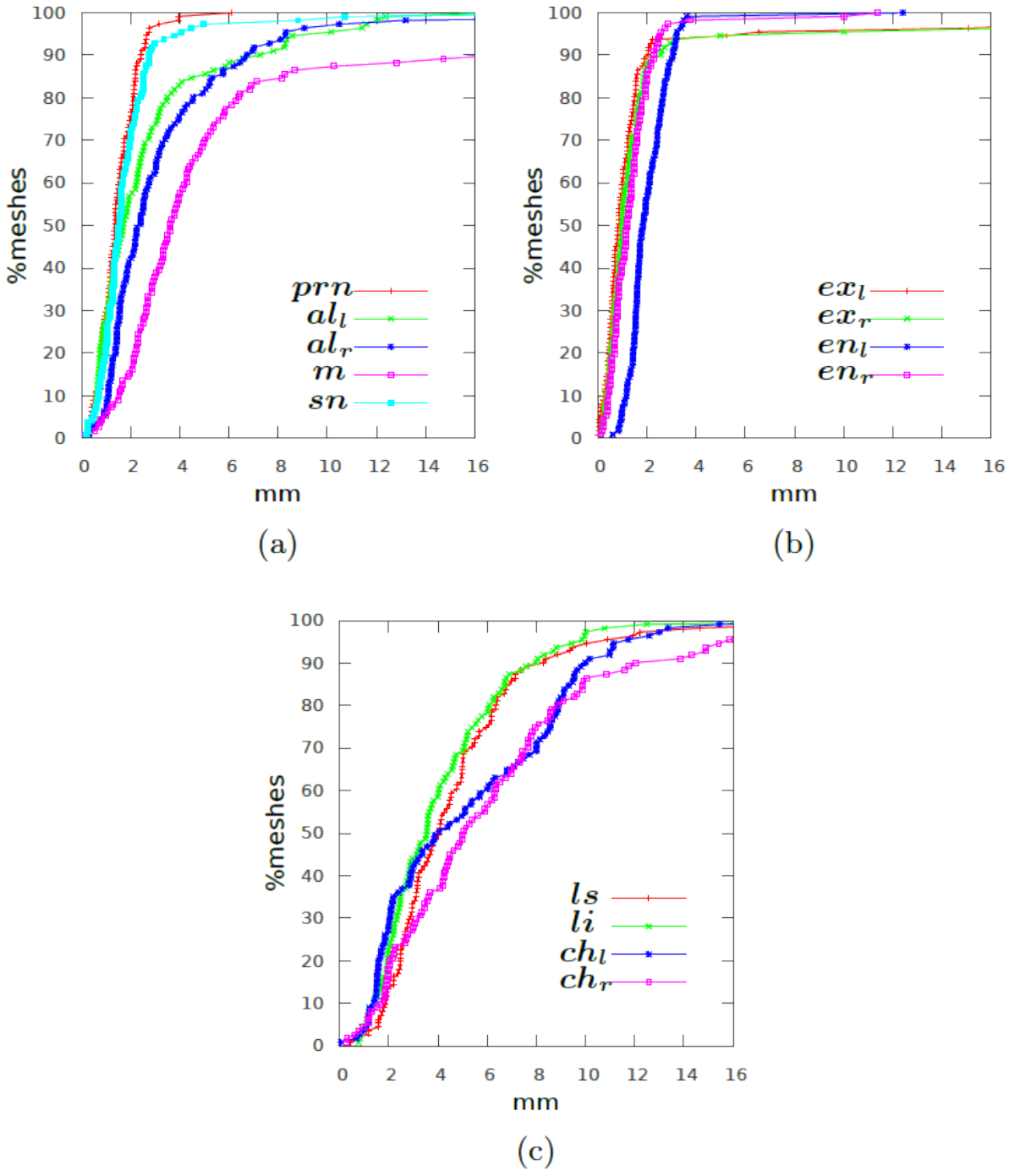


Figure 6.7: Localization accuracy for fiducial points: (a) points in the nose area; (b) the corners of the eyes; (c) points around the mouth.

Chapter 7

Conclusions and Future Work

This thesis presented the development of techniques to deal reliably with datasets that present issue such as size, varying dimensionality, noise. We discussed the importance of reducing the complexity of a problem using a framework that is computationally feasible, adaptable and robust to the noise. Our solution is to identify appropriate scalar fields on the input data, and then to reduce the problem by analyzing its critical points. Our approach deals with the concept of scale to identify relevant structures of the signal at different levels of detail and to produce a framework that is robust with respect to the noise. Our contribution to the classical scale-space approach gives a better and more precise tracking of critical points, by overcoming the issues of previous implementation which gave results that might create inconsistencies in the tracking. We have also shown how concepts coming from topology, such as the topological persistence, can be related to the concept of scale and used in a way analogous to the scale-space filtering to obtain a ranking of importance of critical points of a signal. While the two rankings are related to similar concepts, they actually express different properties of the input signal and can be combined together.

We have shown the application of this techniques to data coming from motion capture systems, an ever growing source of datasets, to identify relevant instants in the human motion, which we then have used to study the synchronization of movements in a person; we obtained strong results that show that synchronization of the events extracted with a multi-scale approach can be used to differentiate between different level of ability in performing a movement.

We also shown , we introduced an ongoing work about motion segmentation, on which we worked on the preprocessing phase to obtain a good input signal for our analysis; due to the lack of adequate dataset available, we developed a tool to gather ground truth on which we ran some analysis to assess the statistical relevance of the gathered data

n conclusion, we shown the usage of our multi-scale approach to extract fiducial points from 3D meshes of human faces; using the surface curvature as our input signal, we can reliably extract

some points that have been shown in the literature to be good candidates for facial recognition. We obtain results that are comparable or better then the state-of-the-art techniques, while also extracting some points that have never been extracted in previous works, to the best of our knowledge.

7.1 Future Works

Once we get a reliable ground truth on our dataset for the segmentation process, we will proceed with the extraction of the segmentation as discussed in Chapter 5 to test our results against the ground truth extracted.

Another possible future work will be focused on the computation of *inter-personal* synchronization: we resample different recordings in order to have all of them with the same length; then, with the usage of the technique of DTW (Dynamic Time Warping), we can find a warping between recordings when the same motion is done with different speeds and rhythm. At this point, relevant instants extracted with our multi-scale approach can be mapped between the two recording, and we can study the synchronization between different performances.

Publications

Referenced in chapter 4:
Paolo Alborno, Nikolas De Giorgis, Antonio Camurri, and Enrico Puppo. 2017. Limbs synchronisation as a measure of movement quality in karate. In Proceedings of the 4th International Conference on Movement Computing (MOCO 2017).
Nikolas De Giorgis, Enrico Puppo, Paolo Alborno and Antonio Camurri. 2018. Evaluating movement quality through intra-personal synchronisation. Submitted for review to IEEE Transactions on Human-Machine Systems.

Referenced in chapter 6:
Nikolas De Giorgis, Luigi Rocca and Enrico Puppo. 2015. Scale-Space Techniques for Fiducial Points Extraction from 3D Faces. In Proceedings of the 18th International Conference on Image Analysis And Processing (ICIAP 2015).

List of Figures

1.1 Examples of critical points: (a) Minimum and maximum of a 1-dimensional function (b) Saddle point of a 1-dimensional function (c) Saddle point on a surface 9

2.1 15

2.2 The correct sequence of flips that leads to the smoothed signal without creating new critical points 16

2.3 In the linear approximation, edge 2 might flip before edge 1, thus creating a new pair of critical points 16

2.4 The linear approximation would cause flip 2 to occur before flip 1; we delay flip 2 and store it, then we proceed to process the subsequent flip 1; since now flip 2 is legal, we can perform it. 17

2.5 A legal sliding of two critical points in two moves: first the flip of edge, which causes a maximum to move on the right; then the flip of edge 2, which causes the minimum to move also to the right, ... 17

2.6 An illegal situation arises when the order of the flips is wrong: flip 2 occur first, but since is a legal move (collapse of two critical points) we let it occur, then when flip 1 is about to get processed, we should avoid it since it creates new critical points, but we can not, since it can not be further delayed, because it is the last flip occurring 18

2.7 An input signal f with its critical points highlighted in red and an initial pairing. . 20

2.8 Steps of persistence computation. Dotted lines are just placeholders to denote that in such interval the function is considered *as if* it were monotonic. Notice that the function is *not* modified by the algorithm, only the relative adjacency of critical points changes. 21

2.9 A 1D signal (a) and its corresponding impulse signals for persistence (b) and lifespan in the scale-space (c). Note that even critical points which are far in the signal can be smoothed out together (those with the same value of persistence). . 24

3.1 An example of how the kinect depth sensors works. In the upper part we can see the infrared speckle pattern. The system is able to differentiate between each possible pattern; it then knows the relative position of the camera and the projector, as well as the angles marked in green; it can then triangulate the position of the point onto which the pattern is projected. 30

3.2 The setup of passive markers and reflective strips (they can be seen on the hand and feet) for a Mocap session 30

4.1 The MoCap skeleton. In red, green, purple and blue the groups of markers used to define the four clusters, summarised in the side table. 35

4.2 From top to bottom: the original velocity extracted from the clusters' barycenter; the smoothed version with a moving average; its derivative, i.e. our acceleration; the persistence values of the acceleration 38

4.3 The same movement, the ending of a punching phase, performed by athletes at different levels (from left to right: level 3, 4 and 5); and the corresponding persistence values: in orange persistence values of the right arm, in green those of the left arm. Movements at higher levels are highly synchronized while the synchronization decreases with the level. The red bracket is the size of the synchronization window $\tau = 40$ used for our analysis, showing that movements of the Level 3 performer will not count as synced, while both Level 4 and Level 5 will contribute positively to the final synchronization index, but with different intensities 39

4.4 Average of the overall synchronisation index Q_τ^{tot} computed on the trials of each level at different values of the synchronisation threshold τ. 42

4.5 The overall synchronisation index for all trials in the dataset. Each bar represents the value of Q_τ^{tot} and is divided into two segments, representing Q_τ^{arms} (lower, lighter) and Q_τ^{legs} (upper, darker), respecitvely. Different hues correspond to the three levels (L3 magenta, L4 cyan, L5 brown). Less and more saturated colours correspond to the two different katas (Heian Yondan lighter; Bassai Dai bolder). Labels inside bars permit to identify different takes by the same subject. 44

4.6 Q-Q plots for the three groups at Levels 3 (a), 4 (b), and 5 (c) support the null hypothesis for each group. 45

4.7 The Gaussians representing the ideal normal distributions of the three groups. The area of overlap below different curves gives the probability that an element is misclassified. Overlap between Level 3 and Level 4 is tiny; while the overlap between Lavel 5 and the other two groups is nearly null. 46

5.1 Markers placed on the head 52

5.2 Markers placed on the back of the torso (on the left) and on the front (on the right) 52

5.3 Markers placed on the back on the arms (on the left), and specifically on the hands (on the right) 52

5.4 Markers placed on the back on the legs (on the left), and specifically on the feet (on the right) 53

6.1 The set of 25 facial landmarks employed by Gupta et al to compute the facial proportions 60

6.2 A saddle surface 62

6.3 The set of 13 fiducial points extracted by our method shown on one of our input meshes 63

6.4 The diagonal scale-space is composed by a sequence of curvature fields, obtained by computing curvature at increasing scales on increasingly smoothed surfaces. . 65

6.5 Maxima (red) and minima (blue) of Gaussian curvature scaled by life (on the left) and strength (on the right). 67

6.6 (a): The first five points and the bounding boxes used to find them. (b): The horizontal line across al_l and al_r divides the face in an upper half and a lower half; the vertical line represents the symmetry axis computed on the given face. (c): Remaining points located through symmetric search, connected by a yellow dashed line, plus bounding boxes for points ls and li. 69

6.7 Localization accuracy for fiducial points: (a) points in the nose area; (b) the corners of the eyes; (c) points around the mouth. 72

Bibliography

[Arg88] Michael Argyle. *Bodily Communication*. Methuen, London, 1988.

[BA00] Charles Beumier and Marc Acheroy. Automatic face verification from 3d and grey level clues. In *11TH PORTUGUESE CONFERENCE ON PATTERN RECOGNITION*, pages 95–101, 2000.

[BC98] R Thomas Boone and Joseph G Cunningham. Children's decoding of emotion in expressive body movement: the development of cue attunement. *Developmental psychology*, 34(5):1007, 1998.

[BR07] Daniel Bernhardt and Peter Robinson. Detecting affect from non-stylised body motions. In Ana C. R. Paiva, Rui Prada, and Rosalind W. Picard, editors, *Affective Computing and Intelligent Interaction*, pages 59–70, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[BT13] Simone Bianco and Francesco Tisato. Karate moves recognition from skeletal motion. pages 86500K–86500K, 2013.

[BWdBP13] Stefano Berretti, Naoufel Werghi, Alberto del Bimbo, and Pietro Pala. Matching 3d face scans using interest points and local histogram descriptors. *Computers & Graphics*, 37(5):509 – 525, 2013.

[BZ13] M. Böckeler and X. Zhou. An efficient 3d facial landmark detection algorithm with haar-like features and anthropometric constraints. In *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, pages 1–8, Sept 2013.

[CCRa$^+$] Cristina Conde, Roberto Cipolla, Licesio J. Rodríguez-aragón, Ángel Serrano, and Enrique Cabello. 3d facial feature location with spin images.

[CLV03] Antonio Camurri, Ingrid Lagerlöf, and Gualtiero Volpe. Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques. *International journal of human-computer studies*, 59(1):213–225, 2003.

[Coh05] L. Cohen. The history of noise [on the 100th anniversary of its birth]. *IEEE Signal Processing Magazine*, 22(6):20–45, Nov 2005.

[Com] Juan Comas. Manual of physical anthropology, volume = 133, number = 3456, pages = 873–874, year = 1961, doi = 10.1126/science.133.3456.873, publisher = American Association for the Advancement of Science, issn = 0036-8075, url = http://science.sciencemag.org/content/133/3456/873.1, eprint = http://science.sciencemag.org/content/133/3456/873.1.full.pdf, journal = Science.

[CWZ+14] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, March 2014.

[DG09] Beatrice De Gelder. Why bodies? twelve reasons for including bodily expressions in affective neuroscience. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1535):3475–3484, 2009.

[DM89] Marco De Meijer. The contribution of general features of body movement to the attribution of emotions. *Journal of Nonverbal behavior*, 13(4):247–268, 1989.

[ea14] D.M. Lane et al. Online statistics education: An interactive multimedia course of study, 2014.

[ELZ00] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 454–, Washington, DC, USA, 2000. IEEE Computer Society.

[Far58] L.G. Farkas. *An attempt to define the attractive face: an anthropometric study*. Raven Press, 1958.

[FM87] L.G. Farkas and I.R. Munro. *Anthropometric facial proportions in medicine*. Thomas, 1987.

[Fri89] Jerome H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[GMB10] Shalini Gupta, Mia K. Markey, and Alan C. Bovik. Anthropometric 3d face recognition. *International Journal of Computer Vision*, 90(3):331–349, Dec 2010.

[GP09] Hatice Gunes and Massimo Piccardi. Automatic temporal segment detection and affect recognition from face and body display. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):64–84, 2009.

[HSE03] C. Hesher, A. Srivastava, and G. Erlebacher. A novel technique for face recognition using range imaging. In *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, volume 2, pages 201–204 vol.2, July 2003.

[JH99] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.

[KBB13] Andrea Kleinsmith and Nadia Bianchi-Berthouze. Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing*, 4(1):15–33, 2013.

[KBBS11] Andrea Kleinsmith, Nadia Bianchi-Berthouze, and Anthony Steed. Automatic recognition of non-acted affective postures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(4):1027–1038, 2011.

[KCV+15] Ksenia Kolykhalova, Antonio Camurri, Gualtiero Völpe, Marcello Sanguineti, Enrico Puppo, and Radosław Niewiadomski. A multimodal dataset for the analysis of movement qualities in karate martial art. In *Intelligent Technologies for Interactive Entertainment (INTETAIN), 2015 7th International Conference on*, pages 74–78. IEEE, 2015.

[KKKM93] Susumu Kuroki, Katsuhiko Kikkawa, Kunihiko Kaneko, and Akifumi Makinouchi. Walkthrough using animation database system move. In Vladimír Mařík, Jiří Lažanský, and Roland R. Wagner, editors, *Database and Expert Systems Applications*, pages 760–765, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[KKVB+05] Asha Kapur, Ajay Kapur, Naznin Virji-Babul, George Tzanetakis, and Peter Driessen. Gesture-based affective computing on motion capture data. *Affective Computing and Intelligent Interaction*, pages 1–7, 2005.

[Koe84] Jan J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370, Aug 1984.

[KVY+16] Dana Kulić, Gentiane Venture, Katsu Yamane, Emel Demircan, Ikuo Mizuuchi, and Katja Mombaur. Anthropomorphic movement analysis and synthesis: a survey of methods and applications. *IEEE Transactions on Robotics*, 32(4):776–795, 2016.

[Lin94] Tony Lindeberg. *Scale-space theory in computer vision*. Kluwer Academic, Boston, 1994.

[LL47] Rudolf Laban and Frederick Charles Lawrence. *Effort*. Macdonald & Evans, 1947.

[LWJ03] Xiaoguang Lu, Yunhong Wang, and A. K. Jain. Combining classifiers for face recognition. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 3, pages III–13–16 vol.3, July 2003.

[MF69] Albert Mehrabian and John T Friar. Encoding of attitude by a seated communicator via posture and position cues. *Journal of Consulting and Clinical Psychology*, 33(3):330, 1969.

[Nav06] W. Navidi. *Statistics for Engineers and Scientists*. McGraw-Hill Higher education. McGraw-Hill, 2006.

[NN07] J. Novatnack and K. Nishino. Scale-dependent 3d geometric features. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct 2007.

[NN08] John Novatnack and Ko Nishino. Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 440–453, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[NNS06] J. Novatnack, K. Nishino, and A. Shokoufandeh. Extracting 3d shape features in discrete scale-space. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 946–953, June 2006.

[PHM03] Frank E Pollick, Joshua G Hale, and Phil McAleer. Visual perception of humanoid movement. 2003.

[PPR10] D. Panozzo, E. Puppo, and L. Rocca. *Efficient multi-scale curvature and crease estimation*, pages 9–16. 2010.

[PPTK13] P. Perakis, G. Passalis, T. Theoharis, and I. A. Kakadiaris. 3d facial landmark detection under large yaw and expression variations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1552–1564, July 2013.

[Pri] Keith Price. Annotated computer vision bibliography.

[QKG02] R Quian Quiroga, T Kreuz, and P Grassberger. Event synchronization: a simple and fast method to measure synchronicity and time delay patterns. *Physical review E*, 66(4):041904, 2002.

[RKG+11] Jan Reininghaus, Natallia Kotava, David Günther, Jens Kasten, Hans Hagen, and Ingrid Hotz. A scale space based persistence measure for critical points in 2d scalar fields. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2045–2052, 2011.

[RP13] Luigi Rocca and Enrico Puppo. *A Virtually Continuous Representation of the Deep Structure of Scale-Space*, pages 522–531. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[SS06] H. Shin and K. Sohn. 3d face recognition with geometrically localized surface shape indexes. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6, Dec 2006.

[SSBQ10] M. Pamplona Segundo, L. Silva, O. R. P. Bellon, and C. C. Queirolo. Automatic face segmentation and facial landmark detection in range images. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(5):1319–1330, Oct 2010.

[Sum09] Kaoru Sumi. Interactive storytelling system using recycle-based story knowledge. In *Proceedings of the 2Nd Joint International Conference on Interactive Digital Storytelling: Interactive Storytelling*, ICIDS '09, pages 74–85, Berlin, Heidelberg, 2009. Springer-Verlag.

[SW65] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 3(52), 1965.

[SWW12] Federico M. Sukno, John L. Waddington, and Paul F. Whelan. 3d facial landmark localization using combinatorial search and shape regression. In *Proceedings of the 12th International Conference on Computer Vision - Volume Part I*, ECCV'12, pages 32–41, Berlin, Heidelberg, 2012. Springer-Verlag.

[Unc16] Aurelio Uncini. *Fundamentals of Adaptive Signal Processing*. Springer Publishing Company, Incorporated, 1st edition, 2016.

[US05] Seiichi Uchida and Hiroaki Sakoe. A survey of elastic matching techniques for handwritten character recognition. *IEICE - Trans. Inf. Syst.*, E88-D(8):1781–1790, August 2005.

[Vas06] Saeed V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, 2006.

[VFC+11] António M VencesBrito, Mário A Rodrigues Ferreira, Nelson Cortes, Orlando Fernandes, and Pedro Pezarat-Correia. Kinematic and electromyographic analyses of a karate punch. *Journal of Electromyography and Kinesiology*, 21(6):1023–1029, 2011.

[vLL74] R. von Laban and F.C. Lawrence. *Effort; economy of human movement*. Number v. 1974,pt. 2 in Effort; Economy of Human Movement. Macdonald & Evans, 1974.

[VR08] Manfred Vieten and Hartmut Riehle. Movement quality of martial art outside kicks. In *ISBS-Conference Proceedings Archive*, volume 1, 2008.

[Wal98] Harald G Wallbott. Bodily expression of emotion. *European journal of social psychology*, 28(6):879–896, 1998.

[Wit83] Andrew P. Witkin. Scale-space filtering. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'83, pages 1019–1022, San Francisco, CA, USA, 1983. Morgan Kaufmann Publishers Inc.

[YTY02] Tomoyoshi Yoshida, Shoichi Takeda, and Sayoko Yamamoto. The application of entrainment to musical ensembles. 2002.